Computer Science Department

TECHNICAL REPORT

A Modular Strategy for Generating Starting Conformations and Data Structures of Polynucleotide Helices for Potential Energy Calculations

Tamar Schlick

Technical Report 348

February 1988

NYU COMPSCI TR-348 c.2
Schlick, Tamar
A modular strategy for
generating starting
conformations and data

NEW YORK UNIVERSITY

Department of Computer Science
Courant Institute of Mathematical Sciences
251 MERCER STREET, NEW YORK, N.Y. 10012

A Modular Strategy for Generating Starting Conformations and Data Structures of Polynucleotide Helices for Potential Energy Calculations

Tamar Schlick

Technical Report 348

February 1988

This work is supported by an NSF Postdoctoral Fellowship in the Mathematical Sciences and by NSF award ASC-8705589.

A Modular Strategy for Generating Starting Conformations and Data Structures of Polynucleotide Helices for Potential Energy Calculations

Keywords: Starting Conformations, Data Structures, Polynucleotide Helices, Potential Energy Calculations

Tamar Schlick

Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, New York 10012

This work is supported by an NSF Postdoctoral Fellowship in the Mathematical Sciences and by NSF award ASC-8705589.

A Modular Strategy for Generating Starting Conformations
and Data Structures of Polynucleotide Helices for
Potential Energy Calculations

Abstract

We describe a simple and rapid algorithm for generating data structures and starting coordinates of polynucleotides for potential energy calculations. The algorithm is tailored to investigations in cartesian coordinate, rather than dihedral angle, space. First, instead of a tree structure for molecular design, we set up a helix from a simple list of bonds for the basic DNA subunits (sugar, phosphate and bases). Second, instead of using successive transformations to obtain a set of coordinates in one reference frame, we apply a simple "matching" routine to patch DNA subunits. Third, we avoid ring closure and geometry optimization by allowing deviations from equilibrium values only for P−O3′ bond lengths and O5′–P–O3′ bond angles at the residue connection sites. A double-stranded helix is constructed from duplex building blocks (2 hydrogen–bonded nucleotides) which are in turn built from the basic structural units. Every building block is constructed from two sets of geometric variables: $\{\alpha, \beta, \gamma, \delta, \chi, P, \tau_{max}\}$, one for each strand. The building blocks are then assembled into a helix by using the 6 rigid body transformations $\{\Delta x, \Delta y, \Delta z, \theta_{ROLL}, \theta_{TILT}, \theta_{TWIST}\}$. For cartesian space programs, generating starting coordinates by this procedure is particularly useful as an alternative to using actual crystal structure coordinates. After describing the algorithm in detail, we illustrate how it was used to generate model A, B, and Z−DNA helices. We conclude by suggesting how the algorithm can be used to pursue a build-up technique and to set up a wide range of starting conformations in the goal of locating novel helical structures.

A Modular Strategy for Generating Starting Conformations
and Data Structures of Polynucleotide Helices for
Potential Energy Calculations

I. Introduction

Most molecular mechanics programs for polymers begin with the generation or the assembly of coordinates for the model. Coordinates can be generated by using mathematical algorithms, read from known crystal structure data, or obtained by computer-aided molecular design. Next, data structures are constructed to supply the program with all the necessary information needed to compute the potential energy of the system. The necessary information consists of atom names, atom types, atomic charges, lists of bonds and bond types, lists of bond and dihedral angles, lists of hydrogen bonds, and lists of excluded atom pairs for the non-bonded interactions (Coulomb and Lennard-Jones). This information is used to assign force constants and energy parameters to each atom, bond, bond angle and dihedral angle sequence. The program then uses this data repeatedly, for energy evaluations in minimization and molecular dynamics calculations.

Several papers have mentioned in general terms approaches for preparing and linking *data structures* of single residues to form protein and nucleic acid models [1-3]. Many other papers have described algorithms for generating *coordinates* for nucleic acid structures when a given set of internal variables (e.g., bond lengths, bond angles and dihedral angles) is given [4-10]. These algorithms generate structures of DNA subunits and then arrange them into helical polymers by using additional helical parameters. The overall strategy is usually that of "constrained model building", since a given set of geometric variables is declared independent and other variables are adjusted so as to generate reasonable helical structures. Often, energy minimization is combined to ensure ring closure for the sugars and optimal geometries at the connection sites along the sugar-phosphate backbone. In some cases, geometry optimization is constrained to maintain, for example, base pair

conformations or global helical parameters.

In these model building procedures, the architecture of the nucleic acid model is generally constructed according to a "tree structure." This structure partitions the atoms of the molecule into categories of main chain, side chain, branch and end atom groups. For example, the DNA backbone atoms constitute the main branch element, the sugar atoms belong to side chains of the main chain, and the bases are branches of the sugars. Once a hierarchical plan for the molecular model is introduced, local coordinates for various segments of the tree structure are determined. Then, each bond vector in the local coordinate system is transformed into a common reference frame by successive applications of rotations and translations [4,11,12].

Now, all these elements of constrained model building – tree structure, successive transformations, and special procedures to guarantee ring closure and reasonable connector geometries – are tailored to the more traditional approach of potential energy investigation in dihedral angle space. As cartesian coordinate studies are becoming increasingly feasible as computing power improves, different approaches can be considered for generating starting conformations and data structures. In fact, generating starting coordinates for potential energy investigations in cartesian coordinate space is an easier task, since special algorithms for ring closure are not required, and only reasonable, rather than perfect, bond lengths and bond angles are sufficient (geometry optimization is combined with energy minimization). Moreover, alternatives to the tree structure and successive coordinate transformations can be considered, as those elements are more natural when the model is built atom-by-atom, rather than unit-by-unit, from an independent set of dihedral angles. Thus, our goal for this work was to devise a simple, rapid, and modular procedure for generating data structures and starting coordinates for DNA helices that is more natural to cartesian space studies. Starting coordinates are particularly useful because many works in cartesian space – for molecular mechanics and dynamics – avoid coordinate generation altogether and use crystal structure coordinates.

First, instead of a tree structure for molecular design, we set up a helix from a simple list of bonds supplied (by the program) for the basic DNA subunits (sugar, phosphate and bases). This bond list suffices to determine all lists of bond angles, dihedral angles, hydrogen bonds, and excluded atom pairs. Second, instead of using successive transformations to obtain a set of coordinates in one common reference frame, we apply a simple "matching" routine to patch DNA subunits. Third, we avoid ring closure and geometry optimization algorithms for the starting helical conformation by allowing deviations from equilibrium values only for P−O3′ bond lengths and O5′−P−O3′ bond angles at the residue connection sites (see Fig. 1).

A double-stranded helix is constructed from duplex dimer building blocks (two hydrogen-bonded nucleotides) which are in turn built from basic structural units. Every nucleotide is built from the following list of geometric parameters: dihedral angles $\{\alpha, \beta, \gamma, \delta, \chi\}$ (see Fig. 1) and sugar pseudorotation variables $\{P, \tau_{\max}\}$ (see ref. [13]). Six degrees of freedom are then available to specify the helical arrangement of these monomers by rigid body transformations: 3 translational and 3 rotational. These helical parameters are assigned as follows. The first two translational variables, Δx and Δy, are used to determine the location of the helix axis, and the third variable Δz describes translation along the helix axis. Two rotations about the long and short axes of the mean base pair plane are closely related to rotations about the x and y axes, respectively. These rotations are known as base pair *roll* and *tilt* ($\theta_{ROLL}, \theta_{TILT}$). The third rotation describes the angle along the helix axis from one base pair to the next and is known as the helical *twist* (θ_{TWIST}). These helical parameters are illustrated in Fig. 2 and will be discussed in more detail later.

By using this building procedure, we can generate reasonable starting coordinates for A, B, and Z-DNA models rapidly by using average geometric data from crystal structures of DNA oligomers. Additionally, any kind of helix can be built when the list of independent variables is enumerated by the user.

Interestingly, many cartesian coordinate programs begin minimization or dynamics calculations from crystal data coordinates. While this might certainly accelerate convergence to a local energy minimum, it might exclude the possibility of finding novel helical structures. The modular procedure to be described here can be used to set up a wide range of starting conformations and to search for helical structures that differ from the *A*, *B*, and *Z*-DNA varieties. Additionally, the algorithm can be used to apply the build-up technique, known from protein applications [14], to nucleic acids. The build-up technique for polymers constructs reasonable starting points for minimization by combining minima of conformational building blocks. Thus, it might be used here to explore conformational space systematically, and perhaps ultimately, to investigate the base sequence/structure relationship.

The algorithm of this article is used in the molecular mechanics program which was described elsewhere [13,15,16]. However, it is general enough to be adopted for other molecular mechanics progams. The algorithmic details will be described as follows. In section II we describe the basic plan used to assemble data structures and coordinates of building blocks to form DNA helices. Without loss of generality we discuss in detail only double-stranded DNA helices; data structures for DNA or RNA single strands can be assembled from nucleotide, rather than duplex dimer, building blocks, and coordinates for single strands can be taken from double strands. In section III we discuss the individual components of the building strategy: how units for the sugar, phosphate, bases and base pairs are set up, and how subroutines for linking units and performing base pair rotations work. In section IV we describe in detail how the building block routine (*BBLOK*) computes data structures and coordinates for every duplex dimer level of the DNA. In section V we illustrate how model *A*, *B*, and *Z*-DNA helices were generated, and in section VI we summarize the major features of the algorithm and mention extensions and applications.

II. The Basic Plan

We begin by giving an overview of the molecular mechanics program and of the central algorithm of this paper, and in subsequent sections we proceed to tackle the details.

A. The Molecular Mechanics Code

In the main (driver) program, the user first specifies whether data structures and coordinates will be read in or generated automatically by the program. If they are to be read in (the integer variable *INFO* is set to 0), the program gets the information from subroutine *GETMOL* (supplied by the user). If the model is a single or double-stranded polynucleotide, this information can be generated automatically by subroutine *SETMOL*. In this case, the user provides a list of variables and arrays describing the model type (single or double strand, RNA or DNA), name, and its base sequence (see Fig. 3). For each nucleotide, the following variables are prescribed: backbone angles $\alpha, \beta, \gamma, \delta$, sugar pucker parameters P and $\tau_{\max}$, and glycosyl rotation χ. Additionally, six helical parameters $-\Delta x, \Delta y, \Delta z$, $\theta_{TILT}, \theta_{ROLL}$ and θ_{TWIST} – are assigned. If the user wishes to generate standard A, B or Z-DNA helices, a helix *type* can alternatively be specified, and the program will read in an appropriate set of all these geometric variables.

In either case that *GETMOL* or *SETMOL* is used to read or generate a molecular model, the data that will become available to the program consist of the following:

	$NAME$	– the model's name (a character variable)
	NA	– number of atoms
	NB	– number of bonds
	NHB	– number of hydrogen bonds
	$IATOM(NA)$	– a character array identifying each atom with a name (e.g., C1′, O2, HC2′)
	$ID(NA)$	– an integer array assigning each atom a type (1 for N, 2 for C, 3 for O, 4 for P, 5 for H)
	$COUL(NA)$	– an array of atomic partial charges
$(B1)$	$IP1(NB)$	– a group of integer arrays providing
	$IP2(NB)$	a list of bonds and associated bond
	$IPT(NB)$	types. Bond i ($i = 1,\ldots,NB$) connects atoms $IP1(i)$ with $IP2(i)$ and is of bond type $IPT(i)$ (see Fig. 4). The bond code is 1 for single, 2 for double, 3 for hydrogen, and 4 for aromatic bonds (see ref. [15] for a discussion of such assignments).
	$XX(NA,3)$	– the coordinate array where $XX(i,j)$ is the jth coordinate ($x, y,$ or z) of atom i.

For illustration, we show in Appendix A a sample input file of *GETMOL* for a deoxycytidine model.

In the next stage of the program, this data is processed to prepare all necessary lists of bond angles, dihedral angles, hydrogen bonds, and exclusion pairs for potential energy calculations. These lists can be calculated from the bond list arrays alone. To faciliate this task, intermediate arrays *NP1(3,NB)*, *NP2(3,NB)* for bond neighbors and *NAT(4,NA)* for atom neighbors are formed (see Fig. 4). Atoms *NP1(j,i)* and *NP2(j,i)* for $j = 1,2,3$ and $i = 1,\ldots,NB$ are the neighbors of atoms *IP1(i)* and *IP2(i)*, respectively; atoms *NAT(j,i)* for $j = 1,2,3,4$ and $i = 1,\ldots,NA$ are the neighbors of atom i. We adopt the convention

that when less than three or four neighbors exist, we enter the nonzero indices first and fill the rest of the lists with zeros. Formation of these lists from the bond list arrays is illustrated in Fig. 4.

Once the bond, bond neighbor, and atom neighbor arrays are available, lists of bond angles can be constructed by taking pairwise combinations of atom neighbors. Similarly, lists of dihedral angles can be computed from pairwise combinations of bond neighbors. Hydrogen bond lists of $\{Donor - Hydrogen - Acceptor\}$ triplets can easily be formulated from the bond and bond neighbor lists: the bond i between the hydrogen and acceptor atom is catalogued as a hydrogen bond type ($IPT(i) = 3$), and the donor in this triplet is the neighbor of the hydrogen atom.

Exclusion lists keep track of atom pairs *(i,j)* which are separated by less than 3 (single, double, or aromatic) bonds. These pairs are usually not included in the non-bonded interactions to avoid double counting energetic contributions from pair and triplet sequences considered already in the bond and bond angle potentials. We form two arrays *IXLUD(16,NA)* and *NXLUD(NA)* for this purpose: for each atom i, all pairs (i, j) where $j = IXLUD(n, i)$ and $n = 1, \ldots, NXLUD(i)$ are not considered in the pairwise non-bonded energy. These lists are formed by excluding for each atom i interactions with its first and second neighbors.

Full details of constructing all these data structure lists, including the FORTRAN code segments, can be found separately [15].

Once the topological information for the molecule has been provided and processed, we transform coordinates to a new basis. In this basis, the first atom is at the origin, the second atom lies on the x-axis, and the third atom lies in the x, y plane. This transformation is used to handle rotation and translation invariance of the energy function. The user then selects one minimization algorithm from 1) a group of library routines for modified Newton, quasi-Newton and nonlinear Conjugate Gradient, or 2) our own version of a truncated

Newton method. Details and references for all these routines can be found in ref. [16]. After a local minimum has been found, control returns to the main program once again, and the geometry and energy of the computed structure are printed.

B. Routine *SETMOL*

The molecular plan for setting up coordinates and data structures for polynucleotide helices is based on a hierarchical partition of elements: units are made into building blocks (BB) and BB's are made into polynucleotides (see Fig. 5). We define a UNIT as a component of a BB. The units include a sugar (SU), phosphate (P), bases (C,G,A,T,U) and base pairs (CG,TA). For each unit, we supply a routine which contains all the data structures outlined in box (B1) and a coordinate generator. The coordinates for the bases are taken from Taylor & Kennard [17] and for the base pairs they are derived from individual base coordinates by aligning partners according to ideal hydrogen bond geometry [18]. The sugar unit is constructed as a function of $\delta, P, \tau_{\max}$ and γ, and the phosphate unit (3 atoms) coordinates are determined from fixed bond lengths and bond angles.

A BB is the next element in our hierarchical plan. For single strands, a BB is a nucleotide (base + sugar + phosphate) and for double strands it consists of a duplex dimer (phosphate + sugar + base pair + sugar + phosphate). For terminal BB's in the model, we replace phosphate units by hydroxyl groups (see Fig. 6).

A BB is formed by linking together units. Each unit has numbered link sites (see Fig. 5) and thus for every link, the units and sites will be specified. A base will be linked to a sugar unit by using χ, and a sugar will be linked to a phosphate unit by using α and β (details in section IV). For now, we consider the case that two units are linked by creating one bond. Since the units are partitioned so that there is no overlap in atoms, the number of atoms and hydrogen bonds for every two linked units is the sum of these numbers for the individual units. Consequently, all data arrays associated with atoms – *IATOM*, *ID*, and *COUL* – are simply concatenated. The number of bonds for two linked units is the sum

of bonds in individual units *plus* one. The new bond lists $IP1$, $IP2$, IPT are updated, after renumbering atoms of $UNIT\ 2$, by concatenating bond lists of $UNIT\ 1$ with the modified bond lists for $UNIT\ 2$ (only $IP1$ and $IP2$ of $UNIT\ 2$ are modified).

The coordinates for two linked units are determined by translating and rotating all atoms of $UNIT\ 2$ so that a proper linked orientation with $UNIT\ 1$ is achieved. Consult Fig. 7. The following procedure is performed:

(a) Attach atom k to $UNIT\ 1$ as fourth in the sequence $i1 - i2 - i3 - k$ by using bond length r_0, bond angle θ_0, and dihedral angle τ_0.

(b) Attach atom $\ell 1$ as fourth in the $i2 - i3 - k - \ell 1$ sequence similarly (use r_1, θ_1, τ_1).

(c) Attach atom $\ell 2$ as fourth in the sequence $i2 - i3 - k - \ell 2$ (use r_2, θ_2, τ_2).

Note: τ_1 and τ_2 are related by the bond angle θ_3 as follows [41]:

$$cos(\tau_1 - \tau_2) = \frac{cos\theta_3 - cos\theta_1 cos\theta_2}{sin\theta_1\ sin\ \theta_2}. \tag{1}$$

For example, when $\theta_1, \theta_2, \theta_3$ are all tetrahedral, $\tau_1 - \tau_2 = 120°$ and when $\theta_1 + \theta_2 + \theta_3 = 360°$, it can be shown that $\tau_1 - \tau_2 = 180°$.

(d) Translate all atoms of $UNIT\ 2$ so that the k atom of $UNIT\ 2$ coincides with the k position constructed in (a).

(e) Rotate all atoms of $UNIT\ 2$ so that the $\ell 1 - k - \ell 2$ sequence of $UNIT\ 2$ coincides with the positions constructed in (b) and (c).

Steps (a) through (c) are standard appending algorithms in coordinate generators and are also described in ref. [15]. Steps (d) and (e) will be performed by the algorithm "MATCH". $MATCH$ is a general-purpose routine that matches 3 given atoms of $UNIT\ 2$ with 3 atoms of $UNIT\ 1$ by performing translation and rotation to $UNIT\ 2$. The prerequisite steps (a), (b), (c) are carried out by appending to $UNIT\ 1$ three atoms $(k, \ell 1, \ell 2)$

at a specified link site with prescribed geometry (we define a routine *APEND3* for this purpose). The resulting coordinates are directed to a temporary 3×3 array which is then used as the *UNIT 1* array in subroutine *MATCH*. Full details will be provided in the following sections.

Such links between units are made for each residue of the helix. That is, subroutine *SETMOL* calls the BB routine *BBLOK* for every base pair level specified in the base sequence. Thus, although the composition and geometry of a particular building block may be repeated, it is clearer to have a loop to construct a BB for each base pair and store it in order, than to include comparisons and copying procedures. Starting from the 5′ end, we call *BBLOK* with the desired base code (for double helices only the base on the left strand is specified). We also supply selected backbone angles and the 6 helical parameters, so that the BB produced at this level has the helix axis as the z axis and is tilted and rolled as desired. The terminal BB's (first and last) will also have hydroxyls, rather than phosphate groups, attached to the sugars (see Fig. 6).

BBLOK produces coordinates for each BB at the same level of the helix (which we call level 0). That is, the base pair lay in the plane $z = 0$ before the tilt and roll rotations were applied. Associated with every BB n $(n = 1, \ldots, NRES)$, we will have all data structures of box (B1) stored in the following corresponding arrays (with an added index for the BB):

$$
\begin{aligned}
& NABB(n), NBBB(n), NHBB(n), \\
& IATMBB(i,n), IDBB(i,n), COULBB(i,n) \quad \text{for} \quad i = 1, \ldots, NABB(n) \\
& IP1(j,n), IP2(j,n), IPT(j,n) \quad \text{for} \quad j = 1, \ldots, NBBB(n) \\
& XBB(i,m,n) \quad \text{for} \quad i = 1, \ldots, NABB(n), \; m = 1,2,3.
\end{aligned}
$$

Additionally, atom index arrays *IBBONE* and *ISRING* will contain the indices for backbone and sugar atoms for each nucleotide of BB n. $IBBONE(i, ichain, n)$ and

$ISRING(j, ichain, n)$ for $i = 1, \ldots, 6$ or $j = 1, \ldots, 5$ correspond to P, O5′, C5′, C4′, C3′, O3′ and O1′, C1′, C2′, C3′, C4′ , respectively. The integer $ICHAIN$ is 1 for the nucleotide on the left and 2 for the one on the right. These arrays are needed for linking the BB's and also for analyzing dihedral angles and sugar pucker parameters later in the program.

The final helix is constructed by translating the BB's along the helix axis (now z-axis) to the proper level (by Δz) and then rotating each BB about the helix axis (by θ_{TWIST}). The nested FORTRAN loop for determining coordinates and data structures for the final model is shown in Fig. 8. We include segments that will generate a helix with a dinucleotide repeat (as the Z form) by allowing 2 alternating helical twists and a shearing procedure. Note that this procedure is general and can be easily extended to model n-nucleotide repeats. Details will be provided in section V when we discuss Z-DNA. The helix handedness (right or left) and stacking arrangement affect the signs of Δz and θ_{TWIST}. For our chosen form of rotation matrix, the right-handed A and B-DNA are stacked above each other ($\Delta z > 0$) with positive helical twists. For the left-handed Z-DNA, base pairs are stacked below each other ($\Delta z < 0$) with positive helical twists also.

To complete the data structure and coordinate assembly for the helix, we update the bond lists by adding inter-residue connections. On the left side we connect atoms C3′ of BB n with atom P of BB $(n+1)$, and on the right side we connect atom O3′ of BB $(n+1)$ with atom P of BB n (see Fig. 6). These atom indices are read from the $IBBONE$ array.

In the next section (III) we describe in more detail coordinate manipulation routines: for each unit, for linking units, and for performing helical rotations on base pairs. In section IV we describe the final data structure and coordinate assembly for each BB as performed in the $BBLOK$ routine.

III. The Components

A. Bases and Base Pairs

Seven units are constructed for bases and base pairs: C, G, T, A, U, CG and TA. The necessary data for each unit, as specified in (B1), are stored in subroutines *UNITC*, *UNITG*, *UNITT*, *UNITA*, *UNITU*, *UNITCG* AND *UNITTA*, respectively. Coordinates for the heavy atoms of the bases are taken on the orthogonal coordinates in the plane $z = 0$ [17]. Appended hydrogens are partitioned in trigonal orientations 1.0 Å away. Coordinates for a hydrogen-bonded base pair are constructed from the base coordinates by using subroutine *MATCH*. For example, to construct coordinates for CG, we append atoms N3, C2, and C4 of cytosine to guanine, and then "overlay" those coordinates with the same atoms in the cystosine unit (see Fig. 6). Ideal hydrogen bond geometries for base pairs are taken from Arnott et al. [18].

For each base, we specify the atom indices for the glycosyl link (N9 for purines and N1 for pyrimidines). Additionally in the base pair units, indices of CG (pyrimidines) and C8 (purines) are stored, as these atoms are used to construct the base pair coordinate axes. The two link sites for base/sugar connections in base pairs are identified as 1 for pyrimidines and 2 for purines.

B. Sugar Unit

The sugar unit was shown in Fig. 5. By including backbone atoms O5′ through O3′ in the sugar unit, we can use the backbone dihedral angle γ for coordinate construction. The specification for the sugar unit routine is

$$UNITSU(PHASE, PUCKAM, DELTA, IDNA, GAMMA),$$

where *PHASE* and *PUCKAM* are P and $\tau_{\max}$; *IDNA* specifies whether the sugar is for DNA or RNA; and *GAMMA* and *DELTA* are the backbone dihedral angles γ, δ. A

value for *DELTA* is optional, because the routine can approximate δ from P and $\tau_{\max}$ (see section V).

Coordinates for the skeletal ring atoms as a function of P and $\tau_{\max}$ are generated using the procedure of Pearlman & Kim [19]. However, since their algorithm essentially fixes τ_{max} at 38.7°, we have modified it by using interpolated polynomials to describe the variations of endocyclic bond angles with τ_{max} (in addition to P). These polynomials interpoalte coefficients in the bond angle expressions from the empirically-derived values of Westhof & Sundaralingam given for τ_{max} = 20°, 30°, 38.7°, and 44° [38]. Exocyclic substituents of the sugar are attached to occupy positions close to tetrahedral [15]. If a value for *DELTA* had been specified, atom C5′ can also be positioned as fourth in the sequence O3′−C3′−C4′−C5′. Atom O5′ is positioned as fourth in the sequence C3′−C4′−C5′−O5′ by using dihedral angle γ.

Three link sites are identified for the sugar unit: site 1 for sugar/base, site 2 for sugar/phosphate at O5′, and site 3 for sugar/phosphate at O3′. Now, in forming a BB, we have a choice to attach a phosphate unit (P,O1,O2) at link site 2 or 3. We chose to attach the phosphate to O5′ (site 2).

C. The Phosphate Unit

To complete our library of units, we define a phosphate unit of 3 atoms: P, O1 and O2. Coordinates for this simple unit of 3 atoms and 2 bonds are built from P−O bond lengths of 1.48 Å and an O2−P−O1 bond angle of 120°. These values are taken from standard values of phosphate diesters [20] as shown in Fig. 9. The phosphate unit has 2 link sites, both stemming from P: link 1 to sugar O5′ atom and link 2 to O3′ atom.

D. Subroutine *MATCH*

Specification: *MATCH(X,NX,K1,L1,M1,Y,NY,K2,L2,M2)*

Purpose: Match (overlay) coordinates of 3 atoms of *UNIT 2* with corresponding atoms of *UNIT 1*. All atoms of *UNIT 2* are translated and rotated so that this match is achieved (see Fig. 10).

Input:

$X(NX,3)$ – coordinate array for *UNIT 1*

$Y(NY,3)$ – coordinate array for *UNIT 2*

$K1, L1, M1$ – 3 indices of atoms in *UNIT 1*

$K2, L2, M2$ – 3 indices of atoms in *UNIT 2* that should be matched with $K1, L1, M1$

Output: Y(NY,3)- new coordinates for UNIT 2

Procedure: 1. Translate all atoms of *UNIT 2* so that $K2$ and $K1$ coincide and are at the origin:

$$S(j) \leftarrow X(K1,j), \quad j = 1,2,3$$
$$Y(i,j) \leftarrow Y(i,j) - Y(K2,j), \quad i = 1,\ldots,NY \text{ and } j = 1,2,3$$
$$X(i,j) \leftarrow X(i,j) - S(j), \quad i = K1, L1, M1 \text{ and } j = 1,2,3$$

2. Construct bond vectors, cross vectors, and matrices of column vectors:

$$X1(j) = X(L1,j), \quad Y1(j) = Y(L2,j), \quad j = 1,2,3$$
$$X2(j) = X(M1,j), \quad Y2(j) = Y(M2,j), \quad j = 1,2,3$$
$$\mathbf{X3} = \mathbf{X1} \times \mathbf{X2}/\|\mathbf{X1} \times \mathbf{X2}\|, \quad \mathbf{Y3} = \mathbf{Y1} \times \mathbf{Y2}/\|\mathbf{Y1} \times \mathbf{Y2}\|$$
$$A = [\mathbf{X1}, \mathbf{X2}, \mathbf{X3}], \quad B = [\mathbf{Y1}, \mathbf{Y2}, \mathbf{Y3}].$$

3. Compute the transformation (rotation) matrix Q so that $QB = A$:

$$Q \leftarrow AB^{-1}$$

4. Compute new coordinates for *UNIT 2* and translate atoms $K1, L1$, and $M1$ of *UNIT 1* to their original positions:

$$\begin{bmatrix} Y(i,1) \\ Y(i,2) \\ Y(i,3) \end{bmatrix} \leftarrow \begin{bmatrix} & & \\ & Q & \\ & & \end{bmatrix} \begin{bmatrix} Y(i,1) \\ Y(i,2) \\ Y(i,3) \end{bmatrix}, \quad i = 1,\ldots,NY$$

$$X(i,j) \leftarrow X(i,j) + S(j), \quad i = K1, L1, M1 \text{ and } j = 1,2,3$$

Remarks: For the bonded atom sequences $\{L1-K1-M1\}$ and $\{L2-K2-M2\}$, the 2 sets of {2 bond lengths, 1 bond angle} must be compatible. This guarantees that a rigid body transformation can be applied to *UNIT 2* to get the proper alignment with *UNIT 1*. For example, for the structures drawn in Fig. 10 in the x, y plane, Q will indicate a 180° rotation in the x, z plane and a 35° rotation in the x, y plane:

$$Q = \begin{bmatrix} -\cos\theta & \sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \theta = 35°.$$

E. "*GROWING*" subroutines: *GROWCG*, *GROWTA*, *GROWSU* and *APEND3*

To facilitate the linking procedure between units, we construct 3 subroutines that "grow" a given unit (i.e. add 3 atoms to it) so that routine *MATCH* can be applied. The idea is to construct a *temporary* array for these 3 coordinates rather than actually add them on to a particular unit. These subroutines will also facilitate the bookkeeping task of maintaining lists of geometric variables that define the link orientation. Thus, when we want to link *UNIT 1* (whose coordinates remain fixed) with *UNIT 2* at a given site, we call a *GROW* routine for *UNIT 1* (let us use the generic name: *GROW1*). *GROW1* will contain all bond lengths, bond and dihedral angles necessary to add 3 atoms at this given site and will produce a 3×3 array $Z(3,3)$. A *MATCH* call can then be made with subroutine arguments $\{Z, 3, 1, 2, 3\}$ corresponding to the *UNIT 1* group $\{X, NX, K1, L1, M1\}$ described above.

We construct the following growing subprograms:

GROWCG	for adding sugar atoms C1′, O1′, C2′ to C, G bases, and CG base pairs;
GROWTA	for adding sugar atoms C1′, O1′, C2′ to T, A bases, and TA base pairs;
and	
GROWSU	for adding base atoms or phosphate atoms at sites 1, 2, *or* 3 (see Fig. 5).

We sketch the structure of a general *GROW* routine which we will denote temporarily by *GROW1*.

Specification: $GROW1(X, NX, IBASET, IGP, LKZ, VAR1, VAR2, ZLGE, ZSML)$.

Purpose: Prepare an array $ZSML(3, 3)$ from a given array $X(NX, 3)$ of $UNIT\ 1$ by linking to $UNIT\ 1$ three atoms at site IGP. The resulting array is used for linking $UNIT\ 1$ with another unit at site IGP.

Input:

$X(NX, 3)$ – coordinate array of $UNIT\ 1$

$IBASET$ – base type (required only for growing base units)

IGP – link site

$VAR1, VAR2$ – 2 dihedral angles that define the linking orientation

$ZLGE(NZ + 3, 3)$ – temporary working array

Output:

LKZ – linker atom of $UNIT\ 1$ at site IGP

$ZSML(3, 3)$ – array with coordinates of the 3 atoms added to $UNIT\ 1$.

Procedure: 1. Direct the action to the appropriate link site and assign appropriate values to indices $I1, I2, I3$ and all geometric variables of Fig. 7.

```
      GOTO (1,2,3), IGP

1     CONTINUE
        LKZ
        I1          } Assign values
        I2          } appropriate for
        I3          } UNIT 1

        BL0
        BL1
        BL2
        BA0
        BA1         } Assign 3 bond lengths,
        BA2         } 3 bond angles and 3 dihedral
        DA0         } angles of Fig. 7. Dihedral
        DA1         } angle assignments will depend
        DA2         } on VAR1 and VAR2, involve
      GOTO 10       } formulas like eq. (1), and differ
                    } from unit to unit, site to site.
2     CONTINUE
        .
        .
        .
      GOTO 10

3     CONTINUE
        .
        .
        .
      GOTO 10

10    CONTINUE
```

2. Append 3 atoms: Atom 1 of $ZSML$ will be positioned as fourth in the sequence $I1-I2-I3-1$ using $BL0, BA0, DA0$. Atoms 2 and 3 will be positioned similarly. We use a simple modular routine called $APEND3$ which takes the linking data prepared above and forms the final 3×3 array Z.

For illustration, we include in Appendix B the subroutine $GROWSU$. At the moment, we only use this routine at site 2 for sugar/phosphate linkages (atoms are appended to the base and not the sugar for sugar/base linkages).

F. Subroutine *BPAXES* ("*B*ase *P*air *AXES*")

Specification: *BPAXES(XBB,NATOMS,IRES,IBASET,DELY,DELX.A,B,C,D, TILT,ROLL).*

Purpose: For a given double-stranded building block, compute reference axes {**E1**, **E2**, **E3**} for the base pair tilt and roll and then apply these rotations. Translate these coordinate axes to a new origin so that the helix axis is the **E3** axis and express the BB coordinates in this new reference frame (see Fig. 2).

Input:

$IRES$ – residue number (BB index)

$IBASET$ – base pair type of this residue : 1(CG), 2(GC), 5(TA), 6(AT), where the first base in the base pair lies on the $5' \rightarrow 3'$ strand with $5'$ at the top (left strand in our convention).

$NATOMS$ – number of atoms in the BB

$XBB(NATOMS, 3, IRES)$ – coordinate array for this BB. The base pair of this BB lies in the plane $z = 0$.

$DELY$ – distance to the helix axis from the C6 (pyrimidine) – C8 (purine) line along the *dyad* (the line perpendicular to the C1′ of pyrimidine sugar – C1′ of purine sugar line). $DELY > 0$ if the helix axis passes in the major groove side (e.g., A – DNA) and < 0 if the axis is in the minor groove side (e.g., Z – DNA).

$DELX$ – displacement of the helix axis from the dyad along the C1′ – C1′ line. $DELX > 0$ if the axis passes on the right side strand and < 0 if it passes on the left side strand.

$A(3)$ – coordinate vector of C1′ on the left strand

$B(3)$ – coordinate vector of C1′ on the right strand

$C(3)$ – coordinate vector of the reference base atom on the left strand (C6 of pyrimidine or C8 of purine)

$D(3)$ – coordinate vector of the reference base atom on the right strand (C8 of purine or C6 of pyrimidine)

$TILT$ – rotation angle about the long axis of the base pair.

$ROLL$ – rotation angle about the short axis of the base pair.

Output: XBB $(NATOMS,\$,IRES)$ – a new coordinate array for this BB with the proper tilt and roll, expressed in the new coordinate system described as follows. **E1** lies parallel to the C1′ (left)–C1′ (right) line; **E2** is perpendicular to **E1** and lies in the base pair plane; **E3** is normal to **E1** and **E2** and forms a right-handed coordinate system. The new origin is placed at a point **F** so that the helix axis is **E3**.

Remark: The C6–C8 line is approximately parallel to **E1**.

Procedure: 1. Compute the new coordinate axes:

$$\mathbf{M} \leftarrow (\mathbf{A}+\mathbf{B})/2$$
$$\mathbf{E1} \leftarrow (\mathbf{B}-\mathbf{A})/\|\mathbf{B}-\mathbf{A}\|$$
$$\mathbf{E3} \leftarrow (\mathbf{B}-\mathbf{A})\times(\mathbf{C}-\mathbf{A})/\|(\mathbf{B}-\mathbf{A})\times(\mathbf{C}-\mathbf{A})\|$$
$$\mathbf{E2} \leftarrow \mathbf{E3}\times\mathbf{E1}$$

2. Solve for s the following system (of unknowns for s and t) describing the intersection of the line from **C** to **D** with **E2**, namely $t\mathbf{C}+(1-t)\mathbf{D} = \mathbf{M} + s\mathbf{E2}$, by applying Cramer's rule [21]:

$$s \leftarrow \frac{det\begin{bmatrix} C(1)-D(1) & M(1)-D(1) \\ C(2)-D(2) & M(2)-D(2) \end{bmatrix}}{det\begin{bmatrix} C(1)-D(1) & -E2(1) \\ C(2)-D(2) & -E2(2) \end{bmatrix}}$$

3. Apply base pair tilt and roll:

$$\mathbf{F} \leftarrow \mathbf{M} + s\mathbf{E2}$$

$$XBB(i,\mathbf{K},IRES) \leftarrow \sum_{j=1}^{3} (XBB(i,j,IRES) - F(j))(\mathbf{EK}(j)),$$
$$\text{for } \mathbf{K} = \mathbf{1},\mathbf{2},\mathbf{3}, \quad i = 1,\ldots,NATOMS.$$

$$\theta_T \leftarrow TILT$$

$$\begin{bmatrix} XBB(i,1,IRES) \\ XBB(i,3,IRES) \end{bmatrix} \leftarrow \begin{bmatrix} cos\ \theta_T & sin\ \theta_T \\ -sin\ \theta_T & cos\ \theta_T \end{bmatrix} \begin{bmatrix} XBB(i,1,IRES) \\ XBB(i,3,IRES) \end{bmatrix}$$
$$\text{for } \ i = 1,\ldots,NATOMS$$

$$\theta_R \leftarrow ROLL$$

$$\begin{bmatrix} XBB(i,2,IRES) \\ XBB(i,3,IRES) \end{bmatrix} \leftarrow \begin{bmatrix} cos\ \theta_R & sin\ \theta_R \\ -sin\ \theta_R & cos\ \theta_R \end{bmatrix} \begin{bmatrix} XBB(i,2,IRES) \\ XBB(i,3,IRES) \end{bmatrix}$$

4. Translate coordinates to a new origin at the helix axis:

$$XBB(i,1,IRES) \leftarrow XBB(i,1,IRES) - DELX$$
$$XBB(i,2,IRES) \leftarrow XBB(i,2,IRES) - DELY$$
$$\text{for } \ i = 1,\ldots,NATOMS$$

IV. The Building Block Routine (BBLOK)

We now complete the description of how coordinates and data structures are assembled to form a BB.

A. BB Coordinates

A nucleotide BB for single strands is constructed by first linking site 1 of the sugar unit to the glycosyl site of the base (sites are shown in Fig. 5). Recall that for linking one unit to another, we need 3 bond lengths, 3 bond angles, and 3 dihedral angles (see Figs. 7, 10). Let us consider the base or base pair unit as *UNIT 1* of Fig. 10. The coordinates of *UNIT 1* remain the same after the linkage and thus the base will be left in the $z = 0$ plane. Consult Table I. For a purine base, we use the N3, C2, N1 atoms to correspond to the $i1, i2, i3$ indices of Fig. 7. We attach C1′ of the sugar (corresponding to k) by using the glycosyl bond length (r_0), bond angle C2−N1−C1′ (θ_0) and dihedral angle N3−C2−N1−C1′ ($\tau_0 = 180°$). Similarly, we attach C1′ and C2′ ($\ell 1$ and $\ell 2$ of *UNIT 2*) by using $r_1, r_2, \theta_1, \theta_2, \tau_1$ and τ_2. For a pyrimidine base, we use N3, C4, N9 triplet for $i1, i2, i3$. All bond lengths and bond angles for base/sugar linkage are variable in crystal structures and some are pucker-dependent. The values we list in Table I include average bond lengths and bond angles (with (*) suffixes) which are approximations from C3′−endo and C2′−endo type sugars [20,22].

Dihedral angle τ_1 is χ: O1′−C1′−N1−C2 (purines) and O1′−C1′−N9−C4 (pyrimidines). Dihedral angle τ_2 can be approximated as $\tau_1 - 180°$ or can be obtained more accurately by using the relation shown in eq. (1). In our program, routine *GROWCG* appends the three sugar atoms by this procedure to base units C,G and base pair unit CG, in preparation for the *MATCH* call. Similarly, *GROWTA* does the task for the remaining base and base pair units.

The second link for creating a nucleotide BB is made between site 1 of the phosphate unit and site 2 of the sugar (O5′). The procedure is also summarized in Table I. Indices

$i1, i2, i3$ of Fig. 7 correspond here to sugar atoms C4′, C5′, O5′ and indices $k, \ell 1, \ell 2$ to P,O1,O2 of the phosphate unit. Note that we use the sugar as *UNIT 1*, the unit whose coordinates remain unchanged, because we wish to preserve the coordinates of the linked base and sugar group, in which the base is oriented in the plane $z = 0$. Backbone angle β (τ_0 of the figure) is used to position P as fourth in the sequence C4′−C5′−O5′−P along with the standrad bond lengths and angles of $r_0 = 1.6$ Å and $\theta_0 = 120°$. To position O1 and O2, we use $r_1 = r_2 = 1.48$ Å, $\theta_1 = \theta_2 = 108°$, and the two dihedral angles τ_1: C5′−O5′−P−O1 and τ_2: C5′−O5′−P−O2. The latter pair of angles is related to the backbone angle α: C5′−O5′−P−O3′, also about the O5′−P bond. Again we can relate τ_1 and τ_2 to α by using eq. (1). By taking bond angle values of 103° for O5′−P−O3′ and 108° for O3′−P−O1 and O3′−P−O2 (see Fig. 9), we obtain the relations $\tau_1 = \alpha - 114°$ and $\tau_2 = \alpha + 114°$. Thus, we have used as variables backbone dihedral angles α and β to link together the sugar and phosphate unit. For illustration, we include in Appendix B routine *GROWSU* which appends these phosphate atoms to the sugar unit at site 2 in preparation for the *MATCH* call.

In summary, in forming the nucleotide BB in this manner, we have used the following variable data: P, $\tau_{\max}$, γ (and optionally δ) to determine coordinates of the sugar unit, χ for sugar/base linkage, and α, β for the sugar/phosphate linkage. When we model a double-stranded polynucleotide, the BB will be a duplex dimer. The procedure for linking is the same as described above (for the nucleotide), but now we link *each* base of the base pair to a sugar and each sugar to a phosphate. Thus we will use two sets of $\{P, \tau_{\max}, \gamma, (\delta), \chi, \alpha, \beta\}$, one for each strand of the double helix. Again we will produce coordinates for a BB in which the bases are in the $Z = 0$ plane. This orientation is convenient for applying the helical transformation of base pair tilt, roll, twist, and translation along the helix axis.

The process just described can be sketched as follows (we describe the "Repeating Motif" in the next section):

1. Initialize counters:

$$NCUM \leftarrow 0 \text{ (cumulative number of atoms)}$$
$$NBCUM \leftarrow 0 \text{ (cumulative number of bonds)}$$
$$NHBCUM \leftarrow 0 \text{ (cumulative number of hydrogen bonds)}$$
$$NLINKS \leftarrow 0 \text{ (cumulative number of inter-residue links)}$$
$$ICHAIN \leftarrow 1 \text{ (strand number)}$$

2. a) Get coordinates of base or base pair for residue N.
   b) Reset counters $NCUM$, $NBCUM$ and $NHBCUM$ to the number of base (or base pair) atoms, bonds and hydrogen bonds.
   c) Enter the base (or base pair) coordinates and data structures to all BB arrays $XBB, IATMBB, IDBB, COULBB, IP1BB, IP2BB, IPTBB$.
3. Link base and sugar: use "motif" with $UNIT\ 1$ = base, $UNIT\ 2$ = sugar.
4. Link sugar with phosphate (unless $\{N$ = first residue, $ICHAIN = 1\}$ or $\{N$ = last residue, $ICHAIN = 2\}$): use "motif" with $UNIT\ 1$ = sugar, $UNIT\ 2$ = phosphate.
5. Update $NCUM$ and $NBCUM$.
6. If a double-stranded model was specified ($IDS = 1$) and $ICHAIN = 1$, then set $ICHAIN \leftarrow 2$ and go to step 3.
7. If N = first or last BB, append 2 terminal hydrogens and add new coordinates, atom labels, atom codes, atomic charges, and hydroxyl links to corresponding BB arrays. (If N = first *and* last BB, 4 terminal groups are involved).
8. Update $NCUM, NBCUM$, and the 3 bond arrays $IP1BB, IP2BB, IPTBB$ by adding all inter-unit links.

The description above shows clearly how coordinates for the base or base pair unit remain unchanged throughout the procedure, and how links are made between two units, rather than between one unit and the BB piece already created.

B. BB Data Structures

The repeating motif mentioned in the previous subsection for linking two units and updating all BB data structures and coordinates proceeds as follows.

Let the following variables and arrays be given:

XBB, $IATMBB$, $IDBB$, $COULBB$ } Filled parts of coordinate and data structure arrays for BB N

$IP1BB, IP2BB, IPTBB$

$UNIT\ 1$ – last unit added to BB N

$NLAST$ – number of atoms in UNIT *1*

$NBLAST$ – number of bonds in UNIT *1*

$NCUM$ – number of atoms in BB N before UNIT *1* was added

$NBCUM$ – number of bonds in BB N before UNIT *1* was added

$NLINKS$ – current number of links made for BB N

$IP1LNK$, $IP2LNK$, $IPTLNK$ } linker bond arrays (currently $NLINKS$ entries are filled)

Note: Linker bonds are not added in the bond arrays of BB N until construction is completed.

We now perform the following sequence of operations:

1. "$GROW$" $UNIT\ 1$ for linking $UNIT\ 2$ at a given site of $UNIT\ 1$. The 3 new coordinates are produced in array $Z(3,3)$.
2. Get coordinates and data structures of $UNIT\ 2$. Call NY, NBY, $Y(NY,3)$, $IATMY(NY)$, $IDY(NY)$, $COULY(NY)$, $IP1Y(NBY)$, $IP2Y(NBY)$, $IPTY(NBY)$ all variables and arrays for $UNIT\ 2$: number of atoms, number of bonds, coordinates, atom labels, atom codes, particle charges and bond lists, respectively.
3. Call the $MATCH$ routine to produce new coordinates for $UNIT\ 2$:

$$CALL\ MATCH(Z,3,1,2,3,Y,NY,K,L,M).$$

Atom indices K, L, and M of $UNIT\ 2$ are those to be matched with coordinate Z atoms 1, 2, and 3, respectively.

4. Update counters and linker bond arrays:

$$
\begin{aligned}
IP1LNK(NLINKS) &\leftarrow (\text{linker atom of } UNIT\ 1) + NCUM \\
NCUM &\leftarrow NCUM + NLAST \\
NBCUM &\leftarrow NBCUM + NBLAST \\
IP2LNK(NLINKS) &\leftarrow (\text{linker atom of } UNIT\ 2) + NCUM \\
NLAST &\leftarrow NY \\
NBLAST &\leftarrow NBY
\end{aligned}
$$

5. Update all coordinates and data structures for the BB by concatenating and, when necessary, also renumbering atom indices:

$$
\begin{aligned}
XBB(NCUM+i,j,N) &\leftarrow Y(i,j), \qquad i=1,\ldots,NY,\ \ j=1,2,3, \\
IATMBB(NCUM+i,N) &\leftarrow IATMY(i) \\
IDBB(NCUM+i,N) &\leftarrow IDY\ (i) \\
COULBB\ (NCUM+i,N) &\leftarrow COULY\ (i)
\end{aligned}
\Bigg\} \quad i=1,\ldots,NY
$$

$$
\begin{aligned}
IP1BB\ (NBCUM+j,N) &\leftarrow IP1Y(j)+NCUM \\
IP2BB\ (NBCUM+j,N) &\leftarrow IP2Y(j)+NCUM \\
IPTBB\ (NBCUM+j,N) &\leftarrow IPTY(j)
\end{aligned}
\Bigg\} \quad j=1,\ldots,NBY
$$

Note that we have started the linking procedure with atom and bond number counters ($NCUM$ and $NBCUM$) that do not count the last unit added to the BB ($UNIT\ 1$).That is necessary if we want to use a general procedure for updated linker lists by adding $NCUM$ into step 4. Only after the first $IP1LNK$ assignment do we update $NCUM$ and $NBCUM$ appropriately. Thus, in effect, we consider $UNIT\ 1$ as part of our BB only after it has been used for linking with another unit.

V. Model A, B, and Z-DNA Helices

In this section we describe how the coordinate and data structure algorithm was employed to generate models of A, B, and Z-DNA helices [20,23-33]. Because these canonical forms possess very distinctive three-dimensional features, it is useful to generate models that contain some general properties (e.g., handedness, sugar pucker) of these structures rapidly. These starting conformations can then be used for subsequent structure analysis by potential energy and molecular dynamics simulations. Polynucleotide conformations were generated by using the same set of geometric variables – $\{\alpha, \beta, \gamma, (\delta), P, \tau_{\max}, \chi\}$ – for each nucleotide, along with one set of helical parameters. The set of geometric variables per residue is taken from averages of analyzed A, B, and Z-DNA crystal structures. The idea of the algorithm shown in Fig. 8 is general enough so that a helix with an n-nucleotide repeat can be constructed also. Thus, we can generate the left-handed dinucleotide repeating Z-DNA from a *single* building block: a CG duplex dimer. This requires, however, estimation of some helical parameters. As far as the author is aware, other coordinate generators for Z-DNA employ actual coordinates for dC and dG residues and generate a helix by applying dinucleotide twists and translations. For the three canonical helices we briefly summarize the data used in the building procedure and mention some of the approximations involved in estimating the sugar puckering parameters. A complete listing of coordinates and data structures can be obtained from the author.

It should be noted at this point that generating model helices from an average set of geometric variables (averaged from crystal data) is not a trivial task. First, variations in geometric quantities – tilt, twist, and roll, backbone, glycosyl and sugar parameters – can be large from one residue to the next. For example, only recently has attention been directed to the variance of $\{P, \tau_{\max}\}$ in crystal structures [34,35]. The studies clearly reveal not only the very wide range of pseudorotation angles and puckering amplitudes in B-DNA, but also the dependence of these computed values on the X-ray refinement method!

Second, even if a helix is built from residues that differ in their individual geometries, unless bond lengths and bond angles are used exactly as found in the crystal, the resulting helix will not be the same as the crystal structure. In particular, bond lengths and bond angles at and around the sugar are variable and pucker-dependent. Thus, the connector geometries along the backbone are highly "vulnerable" to average choices made there. In our case, coordinates for the sugar ring atoms are constructed as a function of P and $\tau_{\max}$, but exocyclic atoms (among them C5′ and O3′) are appended to occupy positions close to tetrahedral.

As we are well aware of these practical model-building difficulties, we did not expect the connector geometries to be ideal. We emphasize again that this was also *not* the goal of the present algorithm. (In fact, it would be an entirely new problem to optimize connector geometries as a function of the helical parameters). The geometric sets described below were chosen after experimentation with several average sets; they were accepted if they produced average connector geometries for O3′–P bond lengths in the range of 1.6 ± 0.05 Å and for O3′–P–O5′ bond angles in the range of $103° \pm 10°$.

A. *A* and *B*-DNA

The *A* and *B*-DNA forms are both right-handed helices. In broad terms, they differ mainly in the location of the helix axis relative to the base point plane, in the tilt of the base pairs, and in the type of sugar pucker. The helix axis runs through the base pair plane in $B-$DNA, and the base pairs are nearly perpendicular to the helix axis. in $A-$DNA on the other hand, the helix axis is displaced from the base pair center toward the major groove side, and the base pairs are tilted from the helix axis. According to Fig. 2, these features corespond to $\{\Delta y \approx 0$ Å, $\theta_{TILT} \approx 0°\}$ for $B-$DNA and $\{\Delta y \approx 4$ Å, $\theta_{TILT} \approx 14°\}$ for $A-$DNA. The sugar pucker type clusters around O1′−endo to C2′−endo for $B-$DNA (roughly, $100° < P < 180°$) and C3′−endo for $A-$DNA. Because of a large observed range of sugar pucker in $B-$DNA, the location of the helix axis, rather than the sugar pucker,

has been suggested as a more reliable criterion for helix characterization [25]. The helical twist in $B-$DNA is about 36° and in $A-$DNA about 33°. This produces about 10 and 11 base pairs per turn in B and $A-$DNA, respectively.

In Table II we list the geometric parameters used to generate model A and B helices, and in Table III we show the resulting connector geometries. Nucleotide parameters were averaged from crystal structures of the following oligomers – dodecamer d(CGCGAATTCGCG) for $B-$DNA [31] and octamer d(GGTATACC) for $A-$DNA [32]. We found that average data sets from X-ray fiber diffraction were less satisfactory for our coordinate generator: connector O3′−P bond lengths can deviate considerably from ideal values of 1.6 Å. Helical parameters $\{\Delta x, \Delta y, \Delta z, \theta_{TILT}, \theta_{ROLL}, \theta_{TWIST}\}$ were estimated for model helices or assembled from the oligomer data mentioned above. Base pair rolls for A and $B-$DNA are quite small and we assigned to them values of 0°.

Crystal structure studies usually provide backbone dihedral angles α through ζ and glycosyl rotation χ. Thus, for our algorithm, we need to estimate sugar variables P and $\tau_{\max}$ [42]. Backbone angle δ: C5′−C4′−C3′−O3′ is correlated to the endocyclic sugar rotation τ_3: O1′−C4′−C3′−C2′ by $\tau_3 \approx \delta - 120°$ (see Fig. 1 for angle definitions). This relationship can be obtained by using eq. (1) twice after introducing an intermediate angle τ: O1′−C4′−C3′−O3′ and estimating 6 bond angles about C4′ and C3′ as tetrahedral.

Once τ_3 is determined, P can be approximated from the pseudorotation equation $\tau_j = \tau_{\max} \cos(P + \frac{4\pi}{5}(j-2))$ for $j = 0, \ldots, 4$ [13] as:

$$P = \pm\left[\cos^{-1}\left(\frac{\tau_3}{\tau_{\max}}\right)\right] - \frac{4\pi}{5}. \qquad (2)$$

As can be seen from this formula, P is not uniquely determined from τ_3 alone. In fact, the sign of the bracketed term is positive when τ_3 is a decreasing function of P and negative when τ_3 is an increasing function of P. Thus, if the range of P were known, the positive

branch of the inverse cosine term should be positive for $0° \leq P < 36°$, $216° \leq P < 360°$, and the negative branch for $36° \leq P < 216°$. In our case, we can use judgment, when using formula (2), for approximating the region of the pseudorotation cycle.

In addition to this ambiguity, a value for the puckering amplitude, $\tau_{\max}$, must be chosen. A range of $38° - 40°$ has been often assumed in nucleic acid studies [36,19], although noticable deviations have been experimentally and theoretically observed [37-39]. Recent papers suggest, however, that accepting constant puckering amplitude is clearly an oversimplification [34,35]. For the A and $B-$DNA helices we tested a range of puckering amplitudes from 20° to 50° and used the relations above to arrive at compatible $\{\delta, P, \tau_{\max}\}$ sets. The set chosen was the one that produced the best overall connector geometries, within our acceptable ranges. For the B-DNA model, we found that our optimal puckering amplitude agrees very closely with the latest crystal refinement [34].

In Figs. 11 and 12 we show stereo views of two dodecamer models: d(GGTATATATACC) for $A-$DNA and d(CGCGAATTCGCG) for $B-$DNA. Views taken perpendicular to the helix axis show the local tilt angle, major and minor grooves. Views down the helix axis show the different arrangement of the base pairs with respect to the helix axis. In $A-$DNA, in contrast to $B-$DNA, a hole down the helix center is formed and the base pairs are arranged along the periphery of the outer helix cylinder.

B. $Z-$DNA

Generating a model of the left-handed $Z-$DNA helix was not as straightforward as A and $B-$DNA because we had to estimate the following parameters:

(1) base pair twist per nucleotide (a twist per dinucleotide repeat of about 60° is usually given),

(2) location of the helix axis – both Δx and Δy of Fig. 2, and

(3) a shearing parameter.

As mentioned previously, coordinate generators that we are aware of construct $Z-$DNA from a dinucleotide repeat. However, theoretically it is possible to generate a $Z-$DNA helix by using rigid body transformations (3 translations and 3 rotations) of one CG duplex dimer. The practical difficulty is choosing the right helical transformations and applying them in the proper order.

The name of the $Z-$DNA helix was originally proposed to describe the zig-zag pattern along the helix backbone. This pattern arises in part from the alternating {*syn* glycosyl rotation, C3′–endo sugar pucker} and {*anti* glycosyl rotation, C2′–endo sugar pucker} observed respectively in guanine and cytosine residues. A peculiar feature of this slender looking helix (12 base pairs per turn) is the stacking pattern (see Fig. 13). For d(GC) steps, we observe the usual base pair stacking where bases on the same strand are stacked vertically. For d(CG) steps, however, we observe a shearing phenomenon: one base pair is "sheared" in relation to the other so that only two individual bases in successive base pairs are stacked on top of each other. Moreover, these stacked basese belong to opposite strands! This arrangement produces an unusual alternating vertical/horizontal path along the polynucleotide backbone.

Now, in order to understand this shearing phenomenon in terms of our coordinate generator, we examined transparent stacking pictures drawn from Z_I coordinates ref. [30]. First, it became clear – by layering these stacking views on top of each other – that the

helix axis of Z_I lies very close to O2 of cytosine. This gave us an estimate for the $\{\Delta x, \Delta y\}$ set of Fig. 2, necessary to position the helix axis in the base pair plane. Second, we realized that the observed shearing is a direct consequence of the asymmetric placement of the helix axis with respect to the short axis of the base pair ($\Delta x \neq 0$). If originally base pairs were stacked directly on top of each other and the helix axis is defined properly for the first base pair, then only every odd base pair will have the helix axis in the right orientation. Thus, when we define a helix axis (by parameters $\{\Delta x, \Delta y\}$) near the cytosine of the first base pair, every *even* base pair will have to be shifted (sheared) by $2\Delta x$. Only then can the helical twist be applied.

Once this procedure was understood, helical twists of 52° for d(GC) steps and 8° for d(CG) steps could be estimated from the stacking pictures. Thus for a d(GC) polymer as shown in Fig. 14, a helical twist of 0° should be applied to the first base pair, 52° to the second, 60° to the third, 112° to the fourth, and so on. This is the formula given in Fig. 8. The same FORTRAN pseudocode segment also shows how the shearing element is incorporated into the helix generator.

All helical parameters used to generate Z_I are summarized in Table II. We note that base pairs are stacked below each other in $Z-$DNA ($\Delta z < 0$) rather than on top of each other, as in A and $B-$DNA. The sign of the helical twists depends on the stacking arrangement, helix handedness, and the choice of rotation matrix (the $sin\ \theta$ and $-sin\ \theta$ entries can be exchanged). In our case the $\{\Delta z < 0$ Å, $\theta_{TWIST} > 0°\}$ combination produces a left-handed $Z-$DNA. Another difference in the $Z-$DNA parameters is the puckering amplitude $\tau_{\max}$ for deoxyguanosines: the puckering amplitude is considerably lower than 39°. The sets of $\{P, \tau_{\max}\}$ were obtained directly from an author of ref. [30].

The stereo views of a model Z_I dodecamer d(CGCGCGCGCGCG) are shown in Fig. 15. A Z_{II} model can be generated similarly, and from end view in ref. [30], location of the helix axis is slightly different. It is interesting to note that during our estimation of the $\{\Delta x, \Delta y\}$ parameters, we saw that very small variations in these sets can produce very

different top views of $Z-$DNA (see Fig. 16). The final set was chosen because it gave good connector geometries (shown in Table III) and resembled very closely the Z_I crystal model [30].

VI. Conclusions

The algorithm presented in this article for generating data structures and starting coordinates for energy minimization has the following features:

(1) It is modular. Double-stranded helices are constructed from duplex dimer building blocks which are in turn built from basic structural units: bases and base pairs, sugars and phosphates.

(2) It is general. Given any base sequence for a single or double-stranded polynucleotide, along with a set of $\{\alpha, \beta, \gamma, P, \tau_{\max}, \chi\}$ per nucleotide and helical parameters, coordinates and data structures can be generated. In particular, any polynucleotide with an n-nucleotide repeat ($n = 1, 2, 3, \ldots$) can be built from a building block of 2 hydrogen-bonded nucleotides. This will require, however, assignments of helical parametrs per nucleotide: n twists and possibly n sets of the other helical parameters $\{\Delta x, \Delta y, \Delta z, \theta_{TILT}, \theta_{ROLL}\}$. The procedure – as we have illustrated for $Z-$DNA – allows a helix axis to be displaced from the base pair dyad. Consequently, an n-periodic shearing pattern is introduced. Thus, the A and $B-$DNA families of helices can be regarded as special cases (with nucleotide repeats and helix axis approximately on the dyad), whereas the $Z-$DNA is representive of a more general helix form.

(3) It is particularly suited for minimization in cartesian coordinate space. A bond list, rather than a tree structure, is used to compute lists of all the necessary geometric sequences for potential energy calculations. Optimization of geometries at inter-residue connections is combined with the second stage of energy minimization.

The algorithm can be extended and applied in many ways. For example, to introduce mismatched base pairs [26] a new base pair unit can be defined. Similarly, new components

such as carcinogens or solvent molecules can be easily added to our library of units. Because we can generate coordinates from any given set of data, the algorithm can also be used to set up various, even hypothetical, kinds of helices to be examined by energy minimization or molecular dynamics simulations. For example, we can investigate a helix with $Z-$DNA segments embedded in $B-$DNA. The possiblity for such structures has been suggested from experimental observation and by model building [28,29]. For helices of this mixed type, as well as for the more standard helices, it may be valuable to pursue the inverse minimization problem: given a fixed building block conformation, what helical parameters will optimize $O3'-P$ bond lengths and $O3'-P-O5'$ bond angles? Solving this problem will be particularly useful for helices like $Z-$DNA where data for the nucleotide conformations are available, but helical parameters per nucleotide are approximated.

Finally, we mention one important application. The algorithm may be useful in applying the build-up technique that has been employed in protein structure investigations to nucleic acids. By starting simulations from combinations of energy minima for individual building blocks, we can attempt to survey systematically the large conformational space of polynucleotides. Important problems include 1) predicting three-dimensional DNA structures from the linear base sequence, 2) computing the observed forms of DNA, and 3) locating novel helical structures. Recent work by Hingerty et al. [40] indicates that prediction of DNA structure from sequence by build-up may indeed be feasible, and we plan to pursue this possibility with the current algorithm.

Acknowledgments

The author wishes to thank Suse Broyde and Charles Peskin for their valuable insights and suggestions.

References

[1] P. K. Weiner and P. A. Kollman, *J. Comp. Chem.* **2**, 287 (1981).

[2] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comp. Chem.* **4**, 187 (1983).

[3] C. Pottle, M. S. Pottle, R. W. Tuttle, R. J. Kinch and H. A. Scheraga, *J. Comp. Chem.* **1**, 46 (1980).

[4] W. K. Olson, in *Topics in Nucleic Acid Structures*, Part 2, S. Neidle, ed., Macmillan Press, London, 1982, pp. 1-79.

[5] A. R. Srinivasan and W. K. Olson, *J. Biomol. Struc. Dyn.* **4**, 895 (1987).

[6] E. R. Taylor and K. J. Miller, *Biopolymers* **23**, 2853 (1984).

[7] K. J. Miller, *Biopolymers* **18**, 959 (1979).

[8] V. B. Zhurkin, Y. P. Lysov and V. I. Ivanov, *Biopolymers* **17**, 377 (1978).

[9] P. De Santis, S. Morosetti, A. Palleschi and M. Savino, *Biopolymers* **20**, 1707 (1981).

[10] H. Sklenar, R. Lavery and B. Pullman, *J. Biomol. Struc. Dyn.* **3**, 967 (1986).

[11] R. A. Scott and H. A. Scheraga, *J. Chem. Phys.* **44**, 3054 (1966).

[12] N. Gō and H. A. Scheraga, *Macromolecules* **3**, 178 (1970).

[13] T. Schlick, C. Peskin, S. Broyde and M. Overton, *J. Comp. Chem.* **8**, 1199 (1987).

[14] M. Pincus, R. Klausner and H. A. Scheraga, *Proc. Natl. Acad. Sci. USA* **79**, 5107 (1982).

[15] T. Schlick, "Modeling and Minimization Techniques for Predicting Three-Dimensional Structures of Large Biological Molecules", Ph.D. Thesis, Courant Institute, Dept. of Mathematics, New York University, October 1987.

[16] T. Schlick and M. Overton, *J. Comp. Chem.* **8**, 1025 (1987).

[17] R. Taylor and O. Kennard, *J. Amer. Chem. Soc.* **104**, 3209 (1982).

[18] S. Arnott, P. J. C. Smith and R. Chandrasekaran, in *Handbook of Biochemistry and Molecular Biology; Nucleic Acids - Volume II*, third edition, G. D. Fasman, ed., CRC Press, Cleveland, Ohio, 1976, pp. 411-422.

[19] D. A. Pearlman and S.-H. Kim, *J. Biomol. Struc. Dyn.* **3**, 85 (1985).

[20] W. Saenger, *Principles of Nucleic Acid Structure*, Springer-Verlag, New York (1984).

[21] G. Strang, *Linear Algebra and its Applications*, second edition, Academic Press, New York, 1980.

[22] S. Arnott and D. W. L. Hukins, *Biochem. J.* **130**, 453 (1972).

[23] A. V. Fratini, M. L. Kopka, H. R. Drew and R. E. Dickerson, *J. Bio. Chem.* **257**, 14686 (1982).

[24] R. E. Dickerson, *Scientific American* **249**, 94 (1983).

[25] R. E. Dickerson, "Usual and Unusual DNA Structures: A Summing Up", in *Unusual DNA Structures*, R. E. Wells and S. C. Harvey, eds., Springer-Verlag, New York, 1987.

[26] G. G. Privé, U. Heinemann, S. Chandrasekaran, L.-S. Kan, M. L. Kopka and R. E. Dickerson, *Science* **238**, 498 (1987).

[27] R. E. Dickerson, B. N. Conner, M. L. Kopka and H. R. Drew, in *Nucleic Acid Research*, K. Mizobuchi, I. Watanabe and J. D. Watson, eds., 1983, pp. 35-59.

[28] A. Rich, A. H. J. Wang and A. Nordheim, in *Nucleic Acid Research*, K. Mizobuchi, I. Watanabe and J. D. Watson, eds., 1983, pp. 11-34.

[29] A. H.-J. Wang, G. J. Quigley, F. J. Kolpak, J. L. Crawford, J. H. van Boom, G. van der Marel and A. Rich, *Nature* **282**, 680 (1979).

[30] A. H.-J. Wang, G. J. Quigley, F. J. Kolpak, G. van der Marel, J. H. van Boom and A. Rich, *Science* **211**, 171 (1981).

[31] R. E. Dickerson and H. R. Drew, *J. Mol. Biol*, **149**, 761 (1981).

[32] Z. Shakked and O. Kennard, in *Structural Biology*, A. McPherson and F. Jurnak, eds., John Wiley & Sons, New York, 1983.

[33] J. D. Watson and F. H. C. Crick, *Nature* **171**, 737 (1953).

[34] E. Westhof, *J. Biomol. Struc. Dyn.* **5**, 581 (1987).

[35] A. E. Garcia and J. A. Krumhansl, *J. Biomol. Struc. Dyn.* **5**, 513 (1987).

[36] W. K. Olson, *J. Amer. Chem. Soc.* **104**, 278 (1982).

[37] P. Murray-Rust and S. Motherwell, *Acta Cryst.* **B 34**, 2534 (1978).

[38] E. Westhof and M. Sundaralingam, *J. Amer. Chem. Soc.* **102**, 1493 (1980).

[39] S. C. Harvey and M. Prabhakaran, *J. Amer. Chem. Soc.* **108**, 6128 (1986).

[40] B. E. Hingerly, S. Figueroa, T. L. Hayden and S. Broyde, "Prediction of DNA Structure from Sequence: A Build-up Technique", submitted paper.

[41] Let us denote bond vectors $i2 - i3$, $i3 - k$, $k - \ell 1$ and $k - \ell 2$ of Fig. 7 by $\mathbf{a}, \mathbf{b}, \mathbf{c}_1$ and $\mathbf{c}_2$, respectively. This relationship between τ_1 and τ_2 can be derived by expressing *cos* τ_1 and *cos* τ_2 in terms of the projections of $\mathbf{a}, \mathbf{c}_1$ and $\mathbf{c}_2$ on $\mathbf{b}$. If we denote those projections by $\mathbf{a}'$, $\mathbf{c}_1'$, and $\mathbf{c}_2'$, respectively, it follows that $\tau_1 - \tau_2 = \phi$ where ϕ is the angle between $\mathbf{c}_1'$ and $\mathbf{c}_2'$. The expression for *cos* ϕ produces eq. (1).

[42] Note that if coordinates for all 5 sugar ring atoms were available, a Fourier series procedure can be used – see ref. [13] above.

Figure Titles

1. Definition of dihedral angles for nucleic acids.
2. Local coordinate system for a base pair.
3. Comments in subroutine *SETMOL*.
4. FORTRAN code segment for *NP1*, *NP2*, and *NAT*.
5. Hierarchical plan for designing helices.
6. Assembly of building blocks to form polynucleotides.
7. Procedure for linking two units.
8. FORTRAN code segment for generating data structures and coordinates for helices.
9. Phosphate unit geometry.
10. Notation of atoms for subroutine *MATCH*.
11. An A–DNA model.
12. A B–DNA model.
13. Stacking patterns in Z–DNA.
14. Shearing pattern in Z–DNA.
15. A Z–DNA model.
16. Top view of various Z–DNA models.

Table Titles

I. Geometric variables used for linking units.

II. Geometric variables used for generating A, B, and Z–DNA models.

III. Connector geometries of the A, B, and Z-DNA models.

FIGURE LEGENDS

Fig. 1. Definition of dihedral angles for nucleic acids. Dihedral angles α , β , γ , δ , ϵ and ζ are defined along the backbone, τ_0 through τ_4 are defined about the endocyclic bonds of the sugar, and χ is defined about the glycosyl bond: O1′–C1′–N9–C4 (purines) and O1′–C1′–N1–C2 (pyrimidines). A dihedral angle A–B–C–D is 0° when all 4 atoms are coplanar and atoms A and D coincide in their projections onto the plane normal to the B–C bond. A positive rotation is defined when the far bond C–D, viewed along the B–C bond, is rotated clockwise with respect to A–B. The figure is based on a drawing in *Principles of Nucleic Acid Structure* by W. Saenger. Copyright c 1984 by Springer-Verlag New York Inc. Used with permission of the publisher and author.

Fig. 2. The local coordinate system for a base pair. A coordinate system must be determined for each base pair residue in order to apply helical transformations (twist, roll, tilt, and translation along the helix axis). Base pair *tilt* and *roll* angles specify rotations about the short and long axis, respectively, of the mean base pair plane. A *twist* is a rotation about the helix axis. These axes are defined as follows. Draw a line from the C1′ (of pyrimidine sugar) to C1′ (of purine sugar). This will be the direction of the long base pair axis. Pass a perpendicular from the midpoint of this line to form the short axis of the base pair, or the *dyad*. The intersection of the dyad with the CG (pyrimidine) –– C8 (purine) line will be the local base pair origin from which **E1** (long axis) and **E2** (dyad) are drawn. A tilt is a rotation about **E2** and is similarly defined positive when it is clockwise as drawn; a positive rotation moves the far base, as viewed along **E1** (guanine here), down below the paper plane. A roll is a rotation about **E2** and is similarly defined positive when it is clockwise as drawn; a positive roll angle moves the far side of the base (major groove),

as viewed along **E2**, down below the paper plane. The **E3** coordinate axis is the normal to the base pair plane which completes a right-handed coordinate system. To position **E3** at the helix axis, we apply translations along **E1** and **E2** (Δx and Δy, respectively). For A and $B-$DNA, the helix axis lies approximately on the dyad ($\Delta x \approx 0$), but for $Z-$DNA the helix axis is displaced from the dyad toward the pyrimidine atom. The $A-$DNA helix axis lies on the major groove side ($\Delta y > 0$) and the $Z-$DNA helix axis lies on the minor groove side ($\Delta y < 0$).

Fig. 3. Comments in subroutine *SETMOL*. The list of variables and arrays that the user provides to set up a molecular model is shown.

Fig. 4. The FORTRAN code segment that computes bond neighbor arrays *NP1*, *NP2*, and atom neighbor array *NAT* from the bond arrays $IP1, IP2, IPT$.

Fig. 5. The hierarchical plan for setting up coordinates and data structures for polynucleotide helices: *units* $\longrightarrow$ *building blocks* $\longrightarrow$ *polynucleotide helices*. The basic units include 5 bases and 2 base pairs – C, G, T, A, U, CG (shown), and TA; a sugar for both deoxyribose (shown) or ribose; and a phosphate. Each unit has numbered linked sites and associated dihedral angles that define the link orientation. Building blocks are *nucleotides* for single strands and 2 hydrogen-bonded nucleotides for double strands. Nucleotides are made by linking a base unit at the glycosyl link site with a sugar at *link 1*; *link 2* of the sugar is then connected to *link 1* of the phosphate. Duplex dimer building blocks are constructed similarly, but both glycosyl sites of the base pairs are used, each for a link with a sugar; each of those sugars is then also linked to a phosphate unit. Polynucleotides are made from the building blocks by applying helical transformations. A single stranded RNA molecule in $A-$form and a DNA double helix in the $B-$form are shown. See also Fig. 6 for a view of the "unrolled" helix and for the building block connection

sites.

Fig. 6. Assembly of building blocks to form polynucleotides. Double-stranded helices are made by stacking duplex dimer building blocks in levels along the helix axis and then applying rotations about the helix axis. The strand on the left runs 5′ ⟶ 3′ and the strand on the right runs 3′ ⟶ 5′ (top to bottom).

Fig. 7. The procedure for linking together two units. Three atoms of *UNIT 2* are appended to *UNIT 1* so that the two units can then be properly aligned. Atoms $\{k, \ell 1, \ell 2\}$ of *UNIT 2* are attached to *UNIT 1* as follows. Atom k is positioned fourth in the $i1 - i2 - i3$ sequence of *UNIT 1* by using bond length r_0, bond angle θ_0, and dihedral angle τ_0. Atoms $\ell 1$ and $\ell 2$ are attached similarly as fourth in sequences $i2 - i3 - k - \ell 1$ and $i2 - i3 - k - \ell 2$ by using $\{r_1, \theta_1, \tau_1\}$ and $\{r_2, \theta_2, \tau_2\}$, respectively. This procedure defines implicitly the bond angle θ_3 in *UNIT 2*. Dihedral angles τ_1 and τ_2 are related to one another by θ_1, θ_2, and θ_3 according to the formula given in the text.

Fig. 8. The FORTRAN loop to generate data structures and coordinates for helices. Building block atoms are translated along the helix axis by Δz ("delz") and rotated about the helix by a helical twist ("angle") for each level. This code shows how a helix with a dinucleotide repeat can be generated by including 2 different helical twists and a shearing parameter (details are discussed in section V for Z–DNA).

Fig. 9. The geometry around the phosphate group. Bond lengths (in Å) and bond angles (in degrees) are approximated from models of phosphate diesters [20]. For the 6 bond angles formed about P, note how backbone O3′–P–O5′ is smaller than tetrahedral (103°) while internal O1–P–O2 is trigonal (120°); other 4 angles are close to tetrahedral values.

Fig. 10. Notation of atoms for subroutine *MATCH*. *MATCH* transforms all *UNIT 2* atoms by rigid body transformations so that atoms *ℓ2–k2–m2* are

aligned with *UNIT 1* atoms *ℓ1–k1–m1*, respectively. If we identify the paper plane as the x, y plane, the transformation here involves a translation in the x, y plane, (to overlay *k2* on *k1*), a 180° rotation in the x, z plane, and a 35° rotation in the x, y plane.

Fig. 11. An A–DNA model. The dodecamer model for A–DNA – d(GGTATATATACC) – is generated with averaged parameters from ref. [32], summarized in Table II. The top picture is a stereo view perpendicular to the helix axis, and the bottom picture is a view down the helix axis. This stereo picture, along with others in this article, is prepared for viewing with stereo glasses. To view with crossed eyes, the left and right images should be interchanged.

Fig. 12. A B–DNA model. The dodecamer model for B-DNA – d(CGCGAATTCGCG) – is generated with averaged parameters from the model of ref. [31], summarized in Table II. The top picture is a stereo view perpendicular to the helix axis, and the bottom picture is a view down the helix axis.

Fig. 13. Stacking diagrams for Z-DNA. A consecutive triplet of 3 base pairs is highlighted in a top view of the generated Z-DNA model to illustrate the different stacking patterns. Looking down the helix axis, we see the darkened base pair on top of the outlined one, and the outlined base pair on top of the dotted one. Stacking in d(GC) steps can be seen in the outlined and dotted couple, while stacking in d(CG) steps can be seen in the darkened and outlined couple. For d(GC) sequences, we observe the normal base pair stacking with a helical twist of about 50°. For d(CG) sequences, however, an unusual stacking pattern can be seen, where cytosines of opposite strands are stacked on top of each other, and the guanines are "sheared". This pattern results for the following two reasons. First, the location of the helix axis is unsymmetric with respect to the short base pair axis; the helix axis (marked with a dot) is shifted toward cytosine (see Figs. 2 and 14 also). Second. only a small helical

twist of about 10° occurs in d(CG) steps.

Fig. 14. Shearing pattern in Z–DNA. This view of an "unrolled" double helix model for Z–DNA shows how bases should be stacked along the helix axis (z axis) so that the helix axis will pass appropriately near the cytosines. The dashed line is the short base pair axis (the *dyad* of A and B–DNA), and Δx shows the displacement from that line to the helix axis. When the coordinate axes are defined appropriately for the first base pair, every even base pair must be "sheared" by $2\Delta x$ to get the proper alignment. Only then can helical twists be applied to produce stacking diagrams as shown in Fig. 13.

Fig. 15. A Z–DNA model. The dodecamer model for Z–DNA – d(CGCGCGCGCGCG) – is generated from averaged parameters of Z_I from ref. [30]. These variables, along with estimated helical parameters, are summarized in Table II. The top picture is a stereo view perpendicular to the helix axis, and the bottom picture is a view down the helix axis. Note how all the cytosines aggregate near the inner part of the helix cylinder with their O2 atoms surrounding the helix axis. The guanine bases, on the other hand, cluster around the outer walls of the helix cylinder.

Fig. 16. Top views of various Z–DNA models. Fig. 16 (a) through 16(d) show how different top views of Z–DNA models can be generated by changing only the helical parameters Δy and Δx (see Fig. 2). The sets corresponding to 16(a) through 16(d) are (−2.9,−2.35), (−3.0, −2.35), (−3.1,−2.35), and (−3.2,−2.35), respectively. They form a vertical path in the base pair plane toward the minor groove. Note how the O2 atoms of cytosine cross each other in 16(a) and are progressively pulled apart as Δy increases in magnitude.

TABLE LEGENDS

Table I. A summary of geometric variables used to link base to sugar and sugar to phosphate units. Atom indices and geometric variables correpond to those drawn in Fig. 7. See the text for a detailed explanation of the linking procedure. All geometric variables involving base/sugar linkages can vary from structure to structure and some are pucker-dependent. The values we list followed by (*) are approximations for C3′−endo and C2′−endo type sugars.

Table II. Geometric variables for model helices. Nucleotide parameters for A and B−DNA were taken from averages of the crystal structures d(GGTATACC) of ref. [32] and d(CGCGAATTCGCG) of ref. [31], respectively. Helical parameters were estimated for model helices or assembled from oligomer data. All angles are given in degrees and all distances in Angstroms. Values for Z−DNA were taken from the Z_I−DNA model of ref. [30]. Parameters $\Delta x, \Delta y$, and the two twists for d(GC) and d(CG) steps were estimated by a procedure described in the text. The sign of Δx depends on the side of the base pair plane (left or right) where cytosine is situated (see Fig. 14).

Table III. Connector geometries for model helices. For each of the generated models, we list the resulting connector geometries between the building blocks. Bond lengths are given in Angstroms and bond angles are given in degrees. The column "STEP" identifies the base step for which the O3′ → P → O5′ connection geometry is described (see Fig. 6). For example, for a C → G step, connector geometry refers to O3′ of cytosine – P of guanine – O5′ of guanine.

APPENDICES

Appendix A. A sample input file of subroutines *GETMOL* for a deoxycytidine model.

Appendix B. Subroutine *GROWSU* for linking a sugar with a phosphate unit. Details of the called routines can be found in ref. [15].

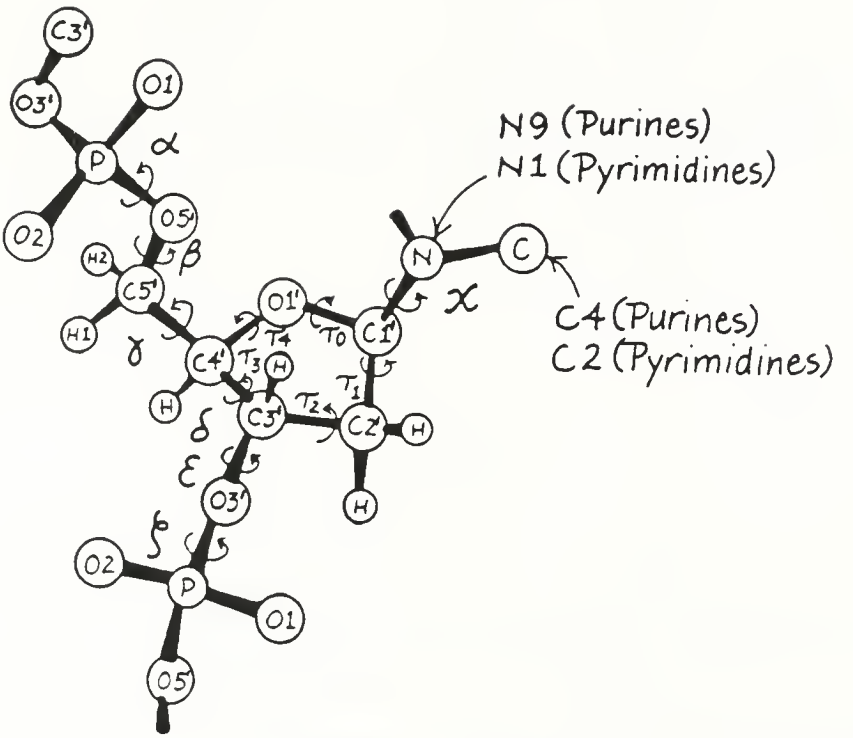

C3'
O3'
O1
P
α
O2
O5'
H2
C5'
β
H1
γ
C4'
O1'
τ4
τ0
C1'
χ
N
C
N9 (Purines)
N1 (Pyrimidines)
C4 (Purines)
C2 (Pyrimidines)
τ3
H
H
τ1
δ
C3'
τ2
C2'
H
ε
O3'
H
ζ
O2
P
O1
O5'

Figure 1

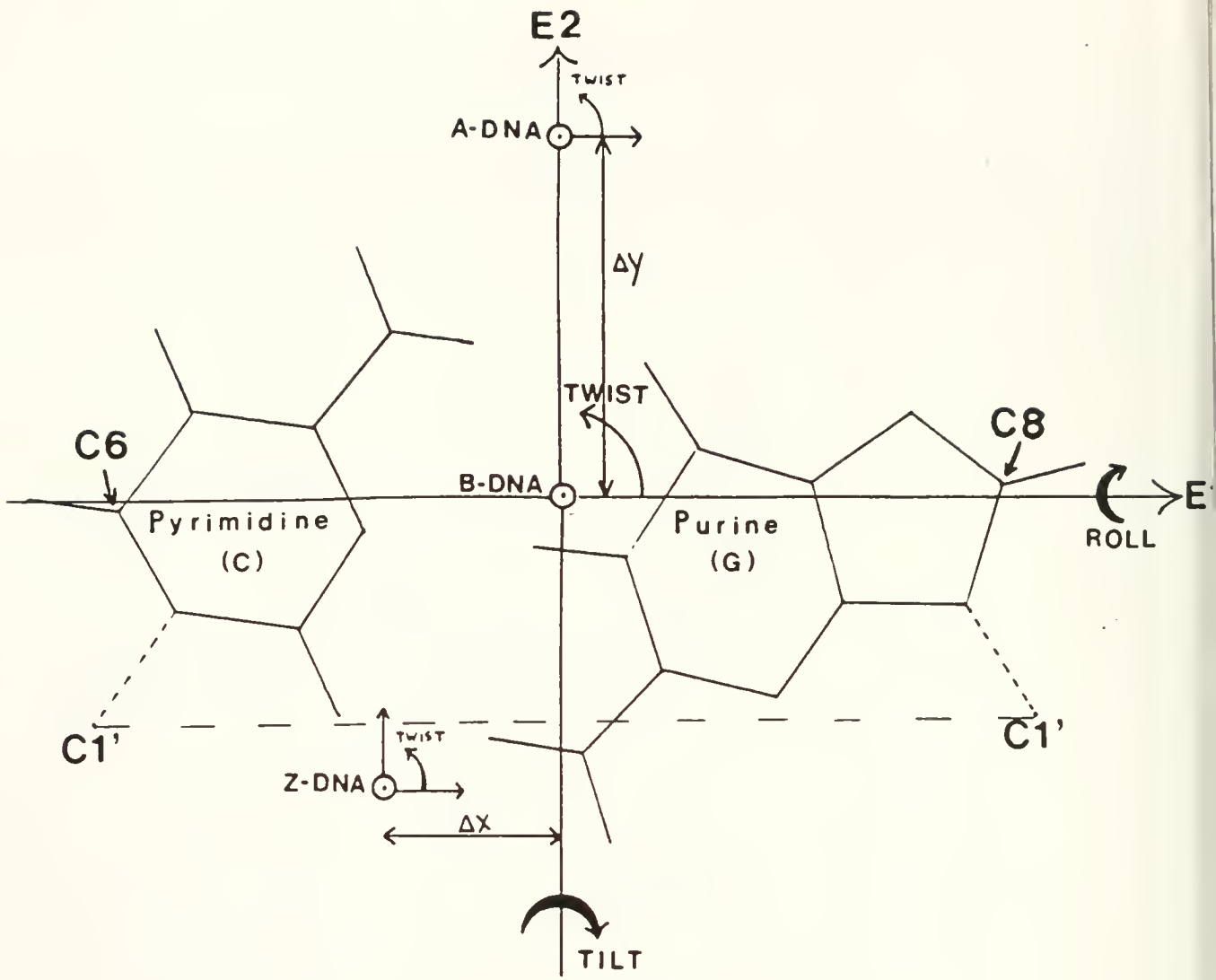

E2
TWIST
A-DNA
ΔY
TWIST
C6
C8
B-DNA
E
Pyrimidine
(C)
Purine
(G)
ROLL
C1'
C1'
TWIST
Z-DNA
ΔX
TILT

Figure 2

```
c ******************************************************
      subroutine setmol
      implicit double precision(a-h,o-z)

c This is the routine that sets up starting conformations for RNA
c and DNA single or double helices.
c A brief description of the variables is given below:
c
c         IDNA      -  1 for DNA
c                      0 for RNA
c         IDS       -  1 for Double Strands
c                      0 for Single Strands
c         NRES      -  # of Residues in the model
c         NAME      -  name of the model
c         DNATYP    -  For each residue, a helix type can be specified
c                      (used for determining dihedral angle values and
c                       sugar puckers for initial conformations):
c                      'A '    for  A-DNA
c                      'B '    for  B-DNA
c                      'ZC'    for  Z-DNA, C Residue
c                      'ZG'    for  Z-DNA, G Residue
c                      'NO'    for  None of the above
c         ISEQ      -  Array for Nucleotide base sequence, starting from
c                      the O5' to O3' end. Only the LEFT strand is
c                      specified for DNA double helices. The base code is:
c
c                      1   C(G)     5   C      NOTE:
c                      2   G(C)     6   G      Codes 1-4 are
c                      3   T(A)     7   T      BASE PAIRS for DNA
c                      4   A(T)     8   A
c                                   9   U
c  For dimensions, we use the following 3 parameters:
c         MAXRES    -  Max # of residues
c         MAXBBA    -  Max # of Building Block (BB) atoms
c         MAXBBB    -  Max # of Building Block (BB) bonds
c
c  Dihedral angles per residue are defined as follows:
c
c         PSEUD     -  Sugar pseudorotation angle
c         TAUM      -  Sugar puckering amplitude
c         CHI       -  Glycosyl rotation:
c                      O1'-C1'-N1 -C2    (C,T)
c                      O1'-C1'-N9 -C4    (G,A)
c         ALFA      -  O3'-P  -O5'-C5'
c         BETA      -  P  -O5'-C5'-C4'
c         GAMA      -  O5'-C5'-C4'-C3'
c        DELTA      -  C5'-C4'-C3'-O3'
c          ETA      -  C4'-C3'-O3'-P
c         ZETA      -  C3'-O3'-P  -O5'
c
c              Suffixes L and R in these angle arrays (PSEUDL(maxres),
c              PSEUDR(maxres), etc.)  stand for  LEFT and
c              RIGHT  sides of the helix. For single strands, only
c              the LEFT side is specified).
c
c  Helical Parameters:
c
c         DELZ      -  Translation along the helix axis per residue
c         DELY      -  Displacement of helix axis from the C8(Purine)-
c                      C6(Pyrimidine) line along the DYAD (the line
c                      perpendicular to C1'(LEFT)-C1'(RIGHT)). DELY > 0
c                      if the axis passes in the major groove (eg. A-DNA)
c                      and < 0 if the axis is in the minor groove.
c         DELX      -  Displacement of helix axis from the DYAD
c                      along the C1'(LEFT)-C1'(RIGHT) line
c                      DELX > 0 if the axis passes on the right side
c                      (the strand running 3' -> 5') and < 0 if it passes
c                      on the left side (5' -> 3' strand).
c         TWIST     -  Twist per residue along the helix axis.
c                      For Z-DNA, we have TWISTG and TWISTC for the twist
c                      from G -> C and C -> G, respectively.
c         TILT      -  Base pair tilt (about the DYAD)
c         ROLL      -  Base pair roll (about the long axis of the base
c                      pair, approx. || to the C8(Purine)-C6(Pyr) line)
c         HELIX     -  Character code for overall helix type, used to
c                      assign helical parameters. Same character codes
c                      as for DNATYP.
c
c  For each BB k, (k = 1, NRES),  the following integers and
c  arrays will be produced:
c
c         NABB(k)   -  # of atoms
c         NBBB(k)   -  # of bonds
c         NHBB(k)   -  # of hydrogen bonds
c         XBB       -  Coordinate array. XBB(i, j, k) is the j'th
c                      coordinate  (j=1,2,3) of atom i  (i=1,NABB(k)).
c         IATMBB    -  Atom label array. IATMBB(i,k) is the label
c                      of atom i  (i=1,NABB(k)).
c         IDBB      -  Atom id array. IDBB(i,k) is the atom code
c                      (1 for N, 2 for C, 3 for O, 4 for P, 5 for N)
c                      of atom i  (i=1,NABB(k)).
c         COULBB    -  Atomic partial charges. COULBB(i,k) is the charge
c                      of atom i  (i=1,NABB(k)).
c IP1BB,IP2BB,IPTBB -  Bond list arrays. Bond j (j=1,NBBB(k)) connects
c                      atom IP1BB(j,k) to IP2BB(j,k) and is of bond type
c                      IPTBB(j,k) (1-single,2-double,3-hydrogen,4-aromatic).
c
c  The final molecular model will have similar parameters
c         NA, NB, NHB, XX, IATOM, ID, COUL, IP1, IP2, IPT, without
c         the last dimension above (which indexes the BB).
c
```

Figure 3

```
c set up the NAT array for  atom neighbors
c-------------------------------------------

c NAT is the array that specifies the neighbors of an atom (see below).
c Thus associated with the atom n  are the 4 (or less)
c neighbors NAT(j,n), where j=1,2,3,4.
c If less than 4 neighbors exist, enter these first, then indicate no
c neigbors by the index 0.
c
c                 NAT(1,n)
c                    |
c                    |
c                    |
c   NAT(2,n) - -  - - n - - - - - NAT(4,n)
c                   /
c                  /
c                 /
c           NAT(3,n)
c
c  The NAT array is filled by searching for every atom in the IP1
c  and then in the IP2 array. When the atom is found, its partner
c  (listed in IP2 and IP1, respectively) is placed
c  in NAT. In the following two loops IA = index of the atom,
c  IX1P1 = index of IP1, and IX1P2 = index of IP2.


      do 18 ia = 1, na
         m=0

c IP1 loop for atom IA:
         do 14 ixip1 = 1, nb
            if ( ip1 (ixip1) .eq. ia ) then
               m = m + 1
               nat ( m, ia ) = ip2 (ixip1)
            endif
   14 continue

c IP2 loop for atom IA:
         do 16 ixip2 = 1, nb
            if ( ip2 (ixip2) .eq. ia ) then
               m = m + 1
               nat ( m, ia ) = ip1 (ixip2)
            endif
   16 continue

c next atom:

   18 continue
```

```
c set up the NP1 and NP2 arrays for  bond neighbors
c-------------------------------------------------

c NP1 and NP2 are arrays that specify the neighbors of the two
c atoms involved in the bonds (see below).
c Thus associated with atom IP1(m) of bond m are the 3 (or less)
c neighbors NP1(j,m), where j=1,2,3
c Similiarly, associated with atom IP2(m) of bond m are the 3 (or less)
c neighbors NP2(j,m), where j=1,2,3.
c If less than 3 neighbors exist, enter these first, then indicate no
c neigbors by the index 0
c
c           NP1(1,m)                          NP2(1,m)
c                \                             /
c                 \                           /
c                  \          bond m         /
c   NP1(2,m) - -  - - IP1(m) - - - - - IP2(m) -  - - -  NP2(2,m)
c                  /                         \
c                 /                           \
c                /                             \
c           NP1(3,m)                          NP2(3,m)
c
c  The array elements of NP1 are collected for every  bond IB from
c  the neighbors of atom  IP1(IB) listed in   NAT( *,IP1(IB))
c  except for  IP2(IB)  (which is the bond partner).
c  Similarly, the array elements of NP2 are collected from NAT(*, IP2(IB))
c  In the following loops IB = index of the bond,
c  IAT = index of atom, and IEX = index of atom bonded to IAT, which is
c  excluded from the NP1 or NP2 array.


      do 24 ib = 1, nb
         m=0

c NP1 loop for bond IB:
         iex = ip2 (ib)
         iat = ip1 (ib)
         do 22 j = 1, 4
            if ( nat (j, iat) .ne. iex ) then
               m = m + 1
               np1 ( m, ib ) = nat (j, iat)
            endif
   22 continue

c next bond:

   24 continue

      do 28 ib = 1, nb
         m=0

c NP2 loop for bond IB:
         iex = ip1 (ib)
         iat = ip2 (ib)
         do 26 j = 1, 4
            if ( nat (j, iat) .ne. iex ) then
               m = m + 1
               np2 ( m, ib ) = nat (j, iat)
            endif
   26 continue

c next bond:

   28 continue
```

Figure 4

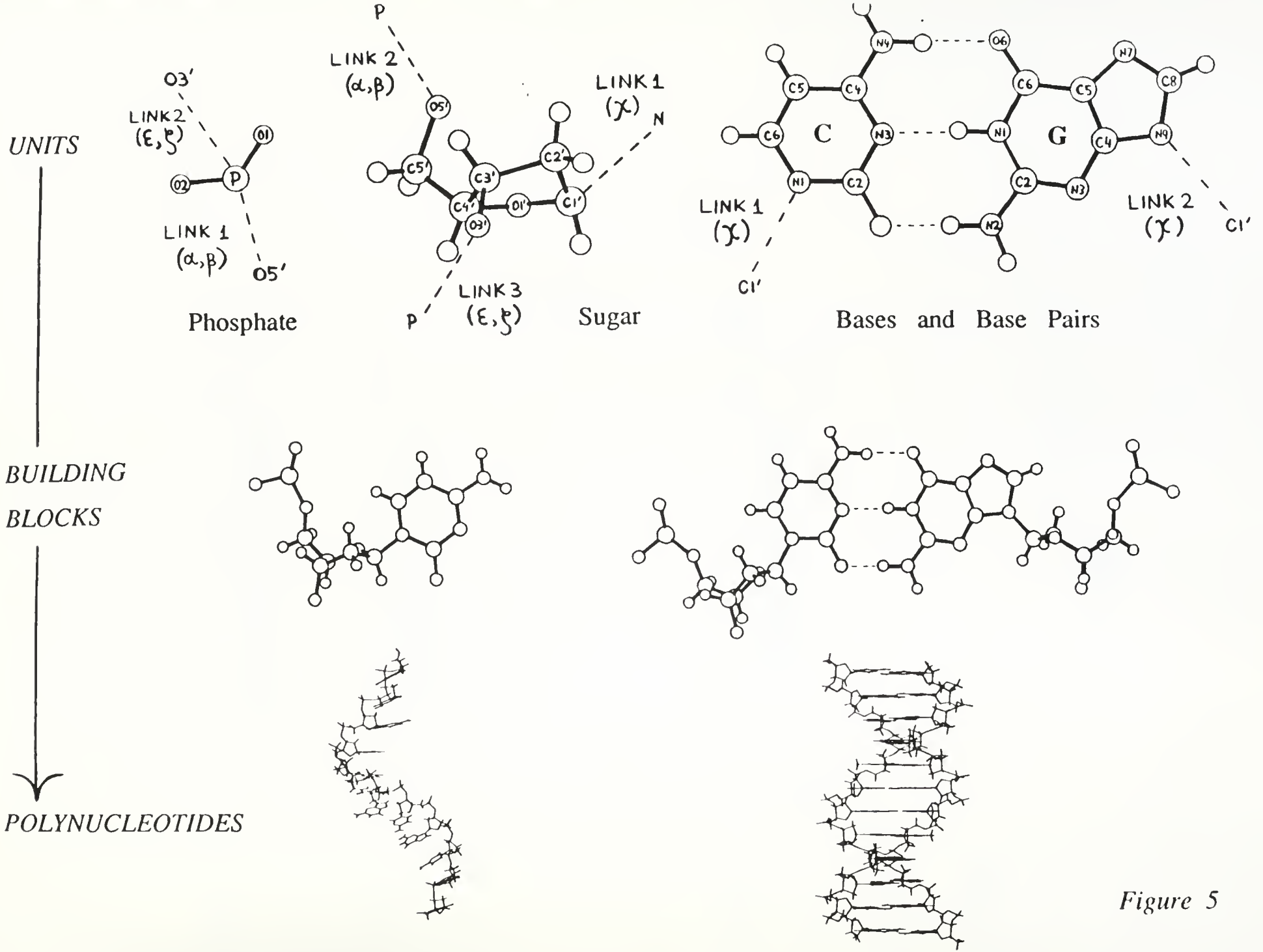

UNITS
O3'
LINK 2
(ε,ζ)
O1
O2
P
LINK 1
(α,β)
O5'
Phosphate
P
LINK 2
(α,β)
O5'
LINK 1
(χ)
N
C5'
C3'
C2'
C4'
O1'
C1'
O3'
P
LINK 3
(ε,ζ)
Sugar
N4
O6
N7
C5
C4
C6
C5
C8
C
N3
N1
G
C6
N1
C2
C2
N3
C4
N9
N2
LINK 1
(χ)
C1'
LINK 2
(χ)
C1'
Bases and Base Pairs
BUILDING
BLOCKS
POLYNUCLEOTIDES

Figure 5

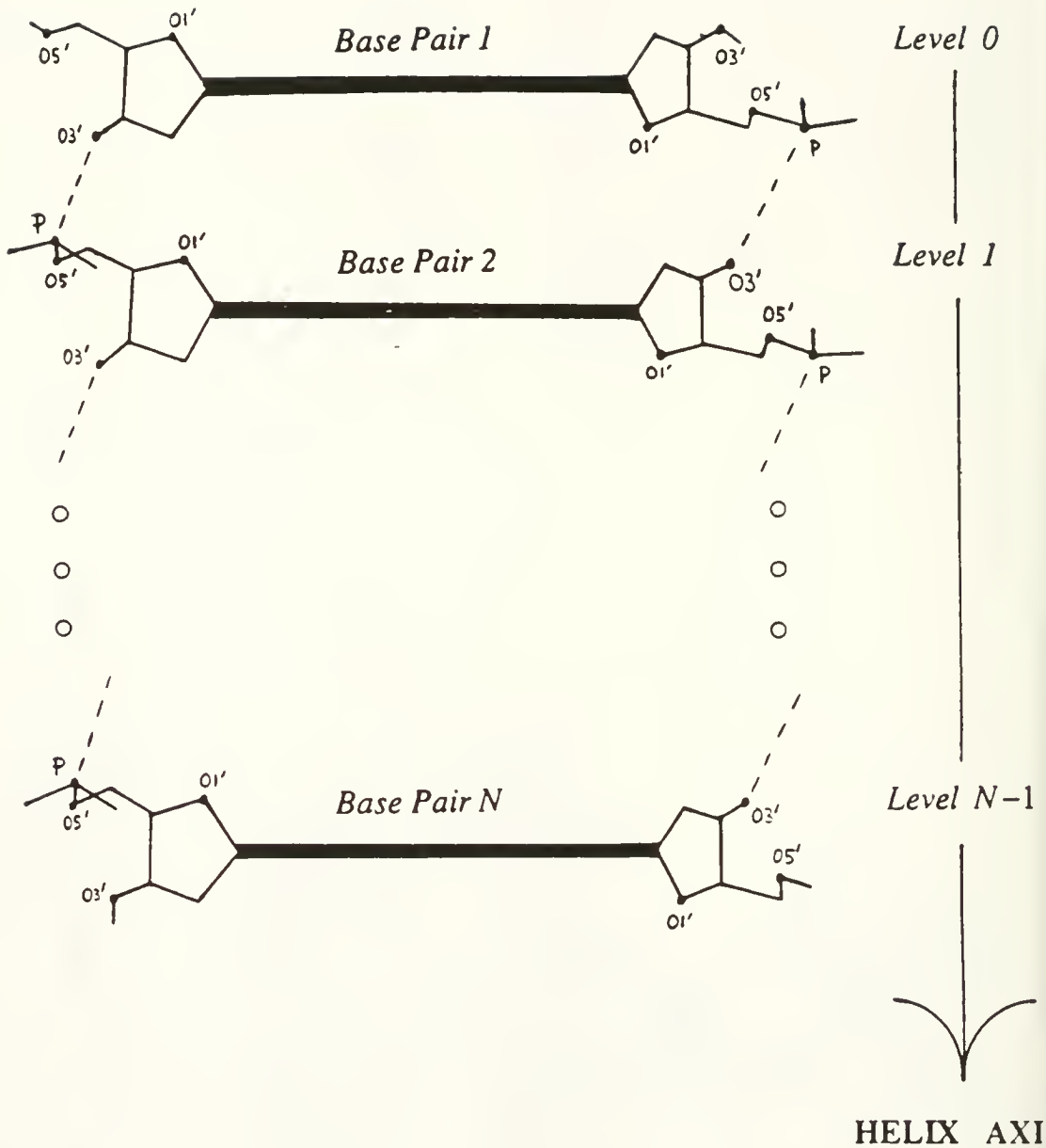

Base Pair 1
Base Pair 2
Base Pair N
Level 0
Level 1
Level N−1
HELIX AXIS
O1′
O5′
O3′
P

Figure 6

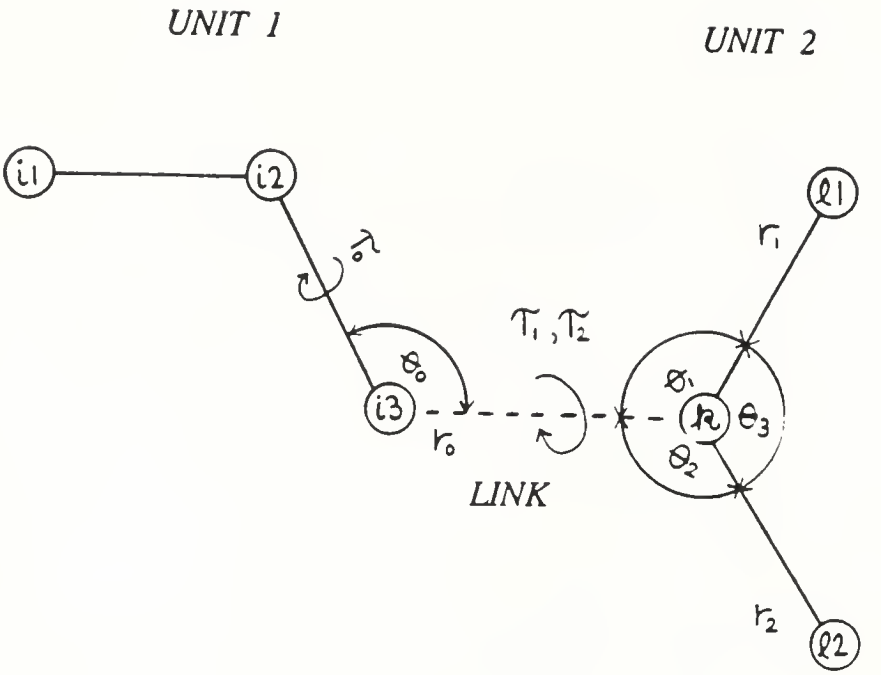

UNIT 1
UNIT 2
i1
i2
i3
k
l1
l2
θ_0
θ_1
θ_2
θ_3
r_0
r_1
r_2
T_1, T_2
LINK

Figure 7

```
   LOOP TO GENERATE DATA STRUCTURES AND COORDINATES FOR HELICES
*****************************************************************

      ncum   = 0
      nbcum  = 0
      nhbcum = 0

      do 100 ires = 1, nres

         level = ires - 1
         trans = dble ( level ) * delz

         if (helix.eq.'Z ') then
            if (dnatyp(ires).eq.'ZC') twist = twstgc
            if (dnatyp(ires).eq.'ZG') twist = twstcg
            angle = (twstcg+twstgc) * int (-level/2)
     #                     + twist * mod (-level,2)
         else
            angle = dble ( - level ) * twist
         endif

         do 50 i = 1, nabb (ires)

            if (helix.eq.'Z ') then
               if ( mod (ires,2) .eq. 0 ) then
                  xbb (i,1,ires) =  xbb (i,1,ires)  + delsx
               endif
            endif

            xx (ncum + i,3) =   xbb (i,3,ires) + trans
            xx (ncum + i,1) =   xbb (i,1,ires) * cosd (angle)
     #                        + xbb (i,2,ires) * sind (angle)
            xx (ncum + i,2) = - xbb (i,1,ires) * sind (angle)
     #                        + xbb (i,2,ires) * cosd (angle)

             coul  (ncum + i ) = coulbb (i,ires)
            iatom  (ncum + i ) = iatmbb (i,ires)
               id  (ncum + i ) = idbb (i,ires)

 50      continue

         do 75 j = 1, nbbb(ires)

            ip1  (nbcum + j ) = ip1bb (j, ires) + ncum
            ip2  (nbcum + j ) = ip2bb (j, ires) + ncum
            ipt  (nbcum + j ) = iptbb (j, ires)

 75      continue

         ncum  = ncum  + nabb (ires)
         nbcum = nbcum + nbbb (ires)
         nhbcum = nhbcum + nhbb (ires)

100   continue

         na =   ncum
         nb =  nbcum
        nhb = nhbcum
```

Figure 8

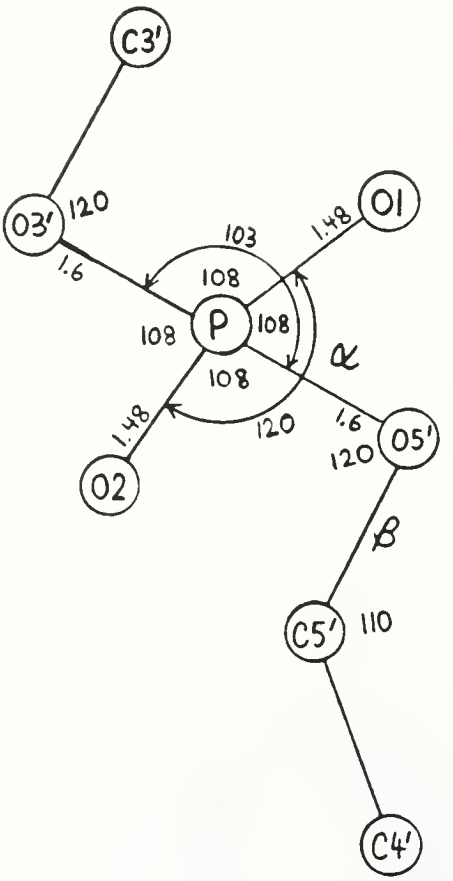

C3'
O3'
120
1.6
103
108
1.48
O1
P
108
108
108
α
1.48
O2
120
1.6
120
O5'
β
C5'
110
C4'

Figure 9

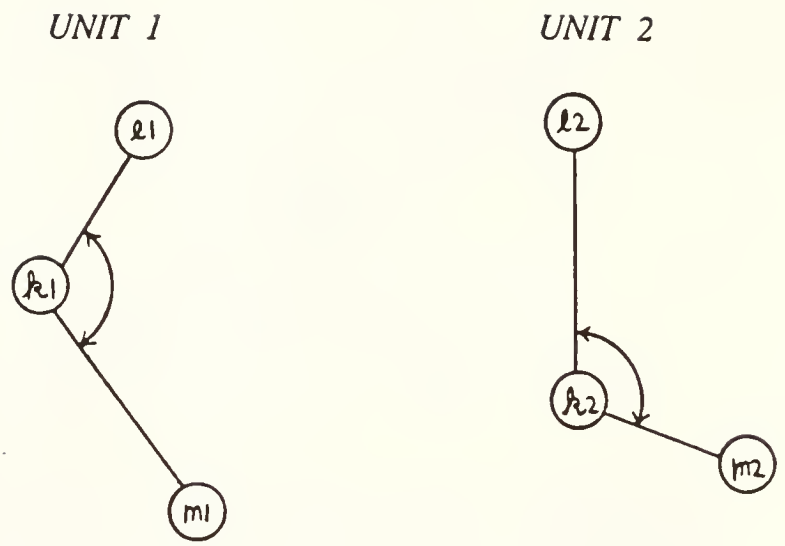


Figure 10

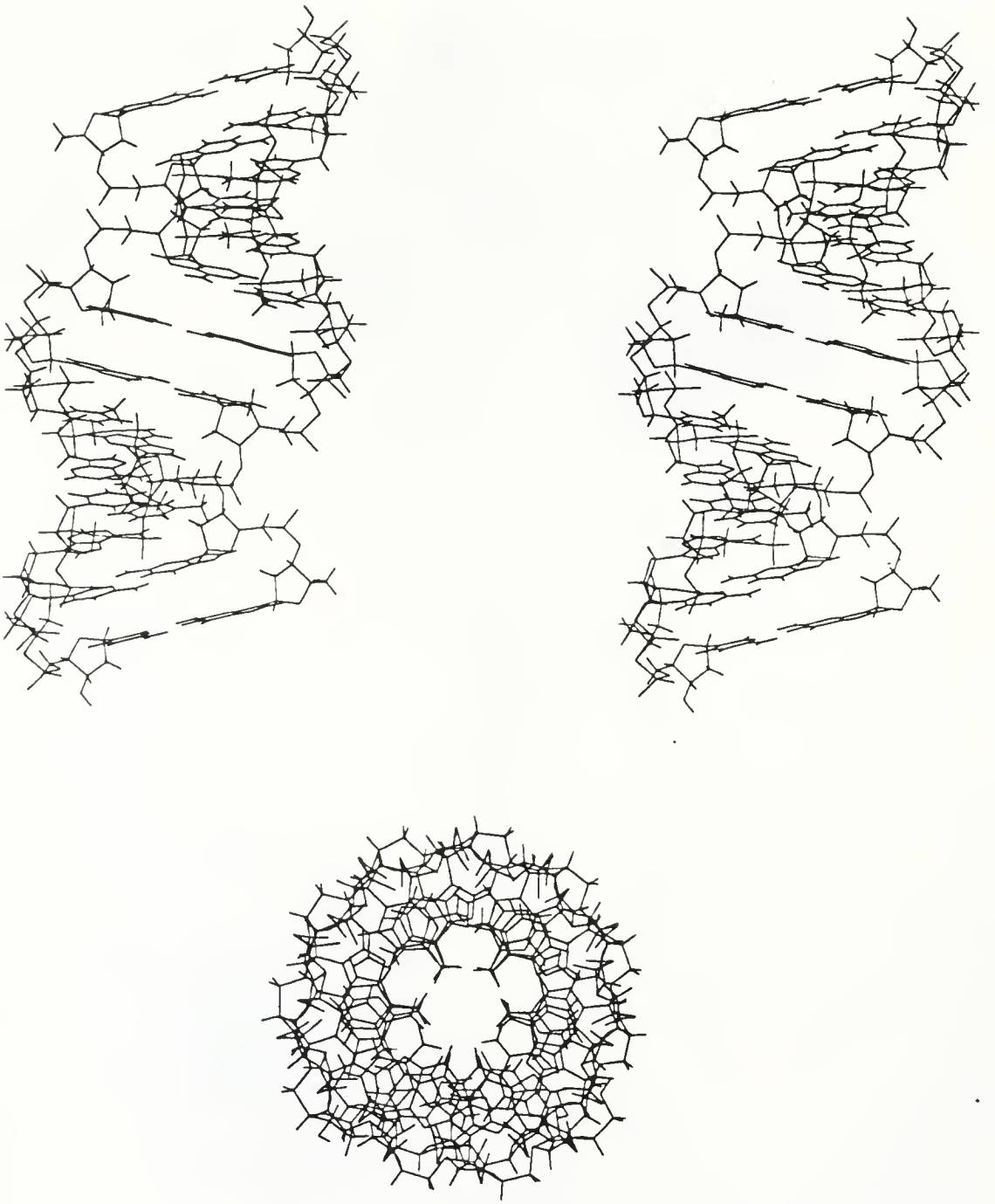


Figure 11

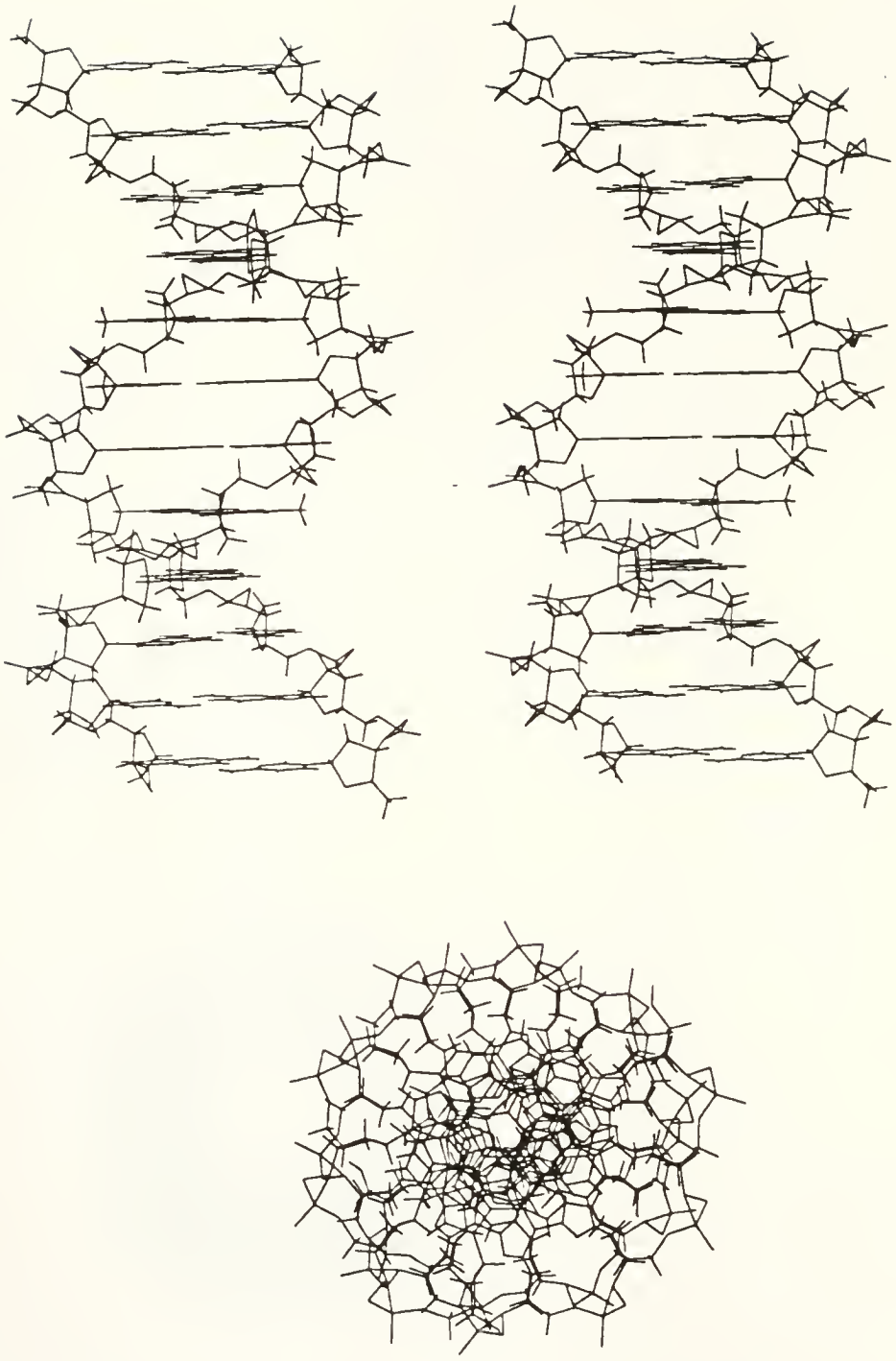


Figure 12

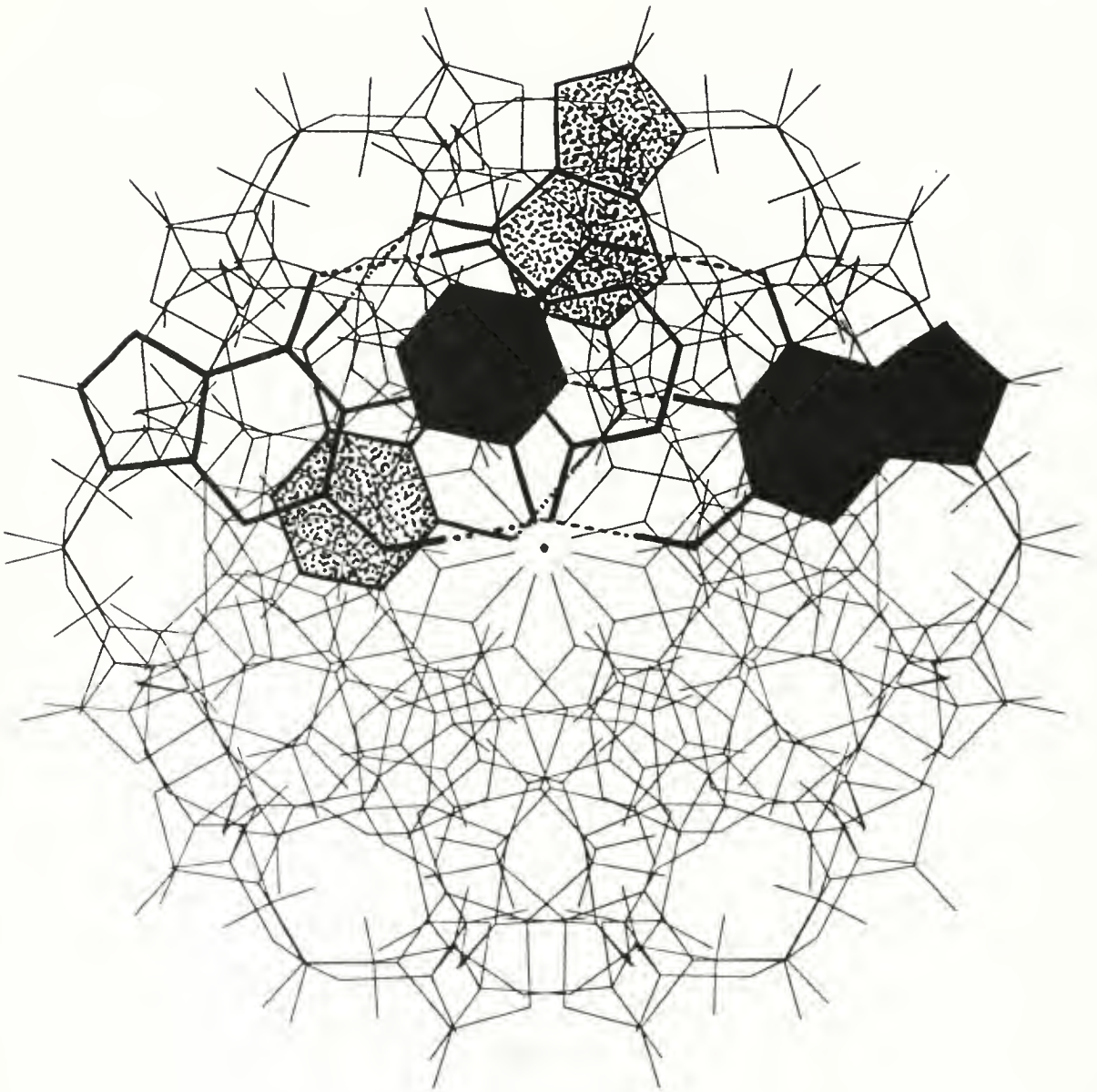


Figure 13

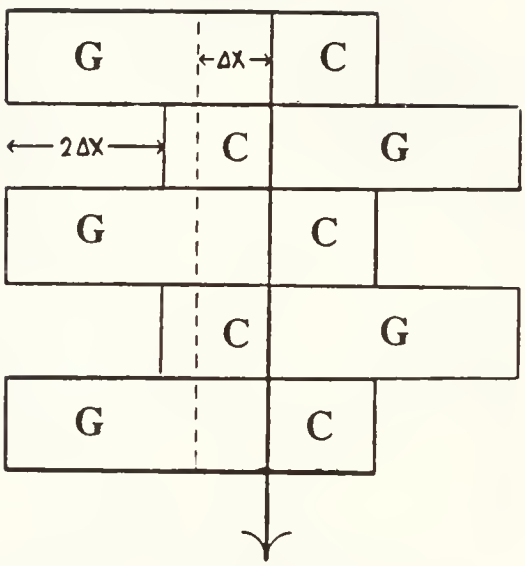


Figure 14

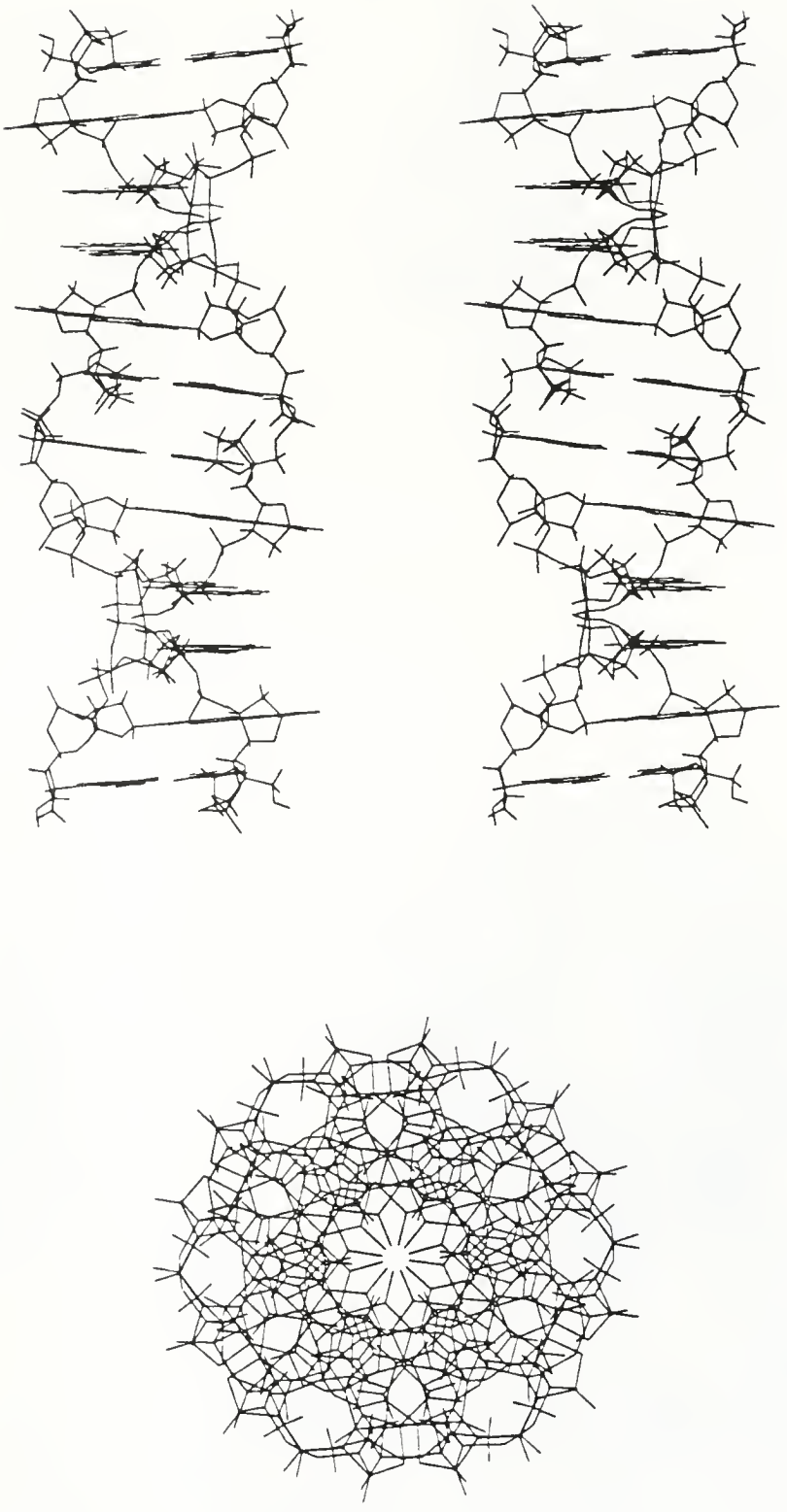
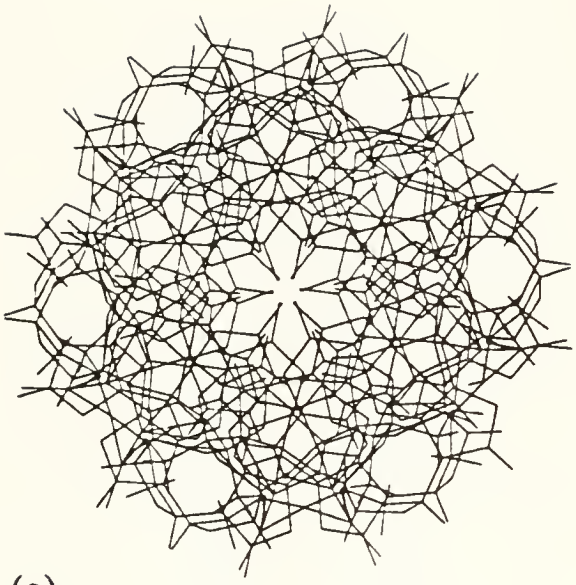
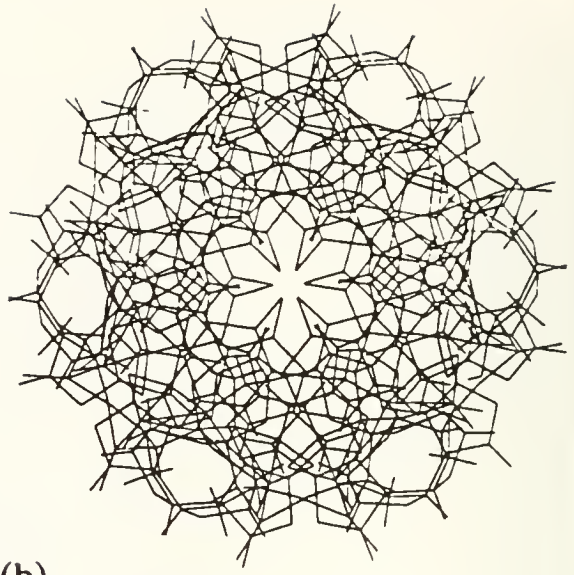


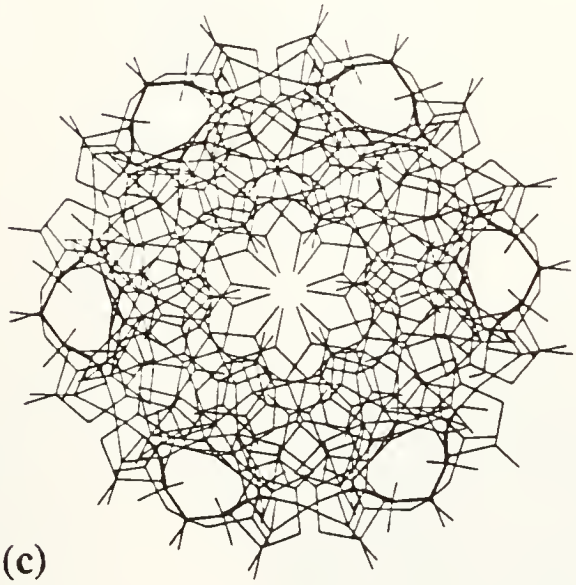
Figure 15



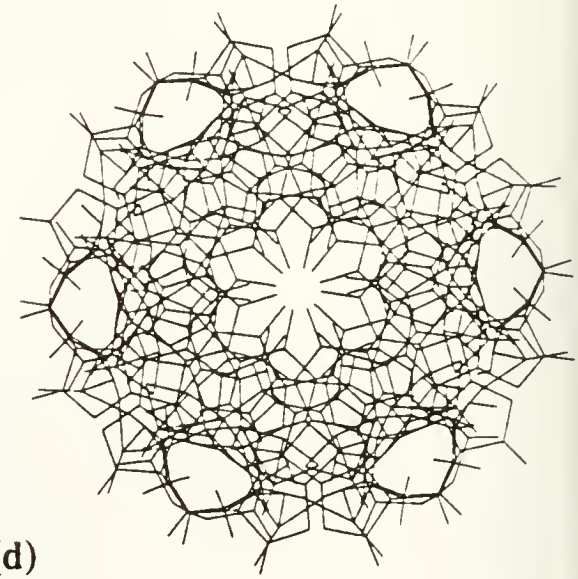
(a)



(b)



(c)



(d)

Figure 16

TABLE I – Building Block Links

			Base (pyrimidine) to Sugar	Base (purine) to Sugar	Sugar to Phosphate
UNIT 1			Base (C,T)	Base (G,A,U)	Sugar
	$i1$		N3	N3	C5′
	$i2$		C2	C4	C5′
	$i3$		N1	N9	O5′
UNIT 2			Sugar	Sugar	Phosphate
	k		C1′	C1′	P
	$\ell 1$		O1′	O1′	O1
	$\ell 2$		O2′	C1′	O2
BOND LENGTHS (Å)	r_0	$i3-k$	1.47	1.47	1.60
	r_1	$k-\ell 1$	1.43	1.43	1.48
	r_2	$k-\ell 2$	1.53	1.53	1.48
BOND ANGLES (°)	θ_0	$i2-i3-k$	118	126	120
	θ_1	$i3-k-\ell 1$	108*	108*	108
	θ_2	$i3-k-\ell 2$	114*	114*	108
	θ_3	$\ell 1-k-\ell 2$	107*	107*	120
					We also use : ϕ : O5′ – P – O3′ = 103 ϕ_1 : O3′ – P – O1 = 108 ϕ_2 : O3′ – P – O2 = 108
DIHEDRAL ANGLES (°)	τ_0	$i1-i2-i3-k$	180	0	β
	τ_1	$i2-i3-k-\ell 1$	χ	χ	$\alpha - \cos^{-1}\left(\frac{\cos\phi_1 - \cos\phi\cos\theta_1}{\sin\phi\sin\theta_1}\right)$
	τ_2	$i2-i3-k-\ell 2$	$\chi - \cos^{-1}\left(\frac{\cos\theta_2 - \cos\theta_1\cos\theta_2}{\sin\theta_1\sin\theta_2}\right)$	$\chi - \cos^{-1}\left(\frac{\cos\theta_2 - \cos\theta_1\cos\theta_2}{\sin\theta_1\sin\theta_2}\right)$	$\alpha + \cos^{-1}\left(\frac{\cos\phi_2 - \cos\phi\cos\theta_2}{\sin\phi\sin\theta_2}\right)$

TABLE II – Geometric Variables for Model Helices

	α	β	γ	δ	P	τ_{max}	χ	Δx	Δy	Δz	θ_{ROLL}	θ_{TILT}	θ_{TWIST}
A-DNA	−62	173	52	88	3	38	−160	0.	4.0	2.87	0.	13.5	32.2
B-DNA	−63	171	54	123	131	36	−117	0.	0.	3.33	0.	0.	37.3
Z-DNA (dG)	47	179	−165	99	−1	23	68	2.5	−3.0	3.72	0.	−7.0	52 (G · C)
Z-DNA (dC)	−137	−139	56	138	152	35	−159	−2.5	−3.0	−3.72	0.	−7.0	8 (C · G)

TABLE III – Connector Geometries for Model Helices

A-DNA d(GGTATATATACC)			*B*-DNA d(CGCGAATTCGCG)			*Z*-DNA d(CGCGCGCGCGCG)		
STEP	O3′ – P	O3′ – P – O5′	STEP	O3′ – P	O3′ – P – O5′	STEP	O3′ – P	O3′ – P – O5′
G → G	1.62	106	C → G	1.61	94	C → G	1.57	94
G → T	1.60	107	G → C	1.57	93	G → C	1.56	103
T → A	1.68	103	G → A	1.53	94			
A → T	1.58	108	A → A	1.55	94			
A → C	1.63	105	A → T	1.68	94			
C → C	1.64	105	T → T	1.63	92			
			T → C	1.55	91			
AVERAGES :	1.63	106		1.59	93		1.57	99

Appendix A

```
c*************************************************************
c
      subroutine getmol
c
c     supply atomic coordinates and data structures for molecule
c
c  THE MODEL USED NOW IS DEOXYCYTIDINE WITH 28 ATOMS:
c  ---------------------------------------------------
c                                          2HN4
c                                            \
c                                            N4--1HN4
c                                           /
c                                  H5      /
c                                   \     /
c                                    C5 ----C4
c                              H6   //     \\
c                               \  //       \\
c                                C6          N3
c                    2H5'  3H5'              /
c                      \  /                 /
c                  1H5'--C5'       O1'      N1 ---- C2
c                          \      /   \     /        \\
c                           \    /     \   /          \\
c                            C4'        C1'            O2
c                           /  \H3' 2H2'/  \
c                      H4'     \|    |/    H1'
c                              C3'---C2'
c                                  /     \
c                                 /       \
c                            H3'-O3'       1H2'
c
c  ---------------------------------------------------
      implicit double precision(a-h,o-z)
      parameter(maxb=1200)
      parameter(maxa=1200)
      parameter(maxhb=100)
      common/sizes/na,nb,nhb
      character iatom(maxa)*4
      common/coord/xx(maxa,3),iatom,id(maxa)
      common/graph1/ipl(maxb),ip2(maxb),ipt(maxb)
      character name*50
      common/nonbon/1ja,1jb,coul
      double precision 1ja(5,5),1jb(5,5),coul(maxa)
      double precision tet,trig,chi,phase,puckam,delta
      integer idna, inring(5),inexo(5,2)

      tet=109 4712206344907d0
      trig=120.0d0
      chi=315.0d0

c NUMBER OF ATOMS, BONDS AND HYDROGEN BONDS

      name = 'DEOXYCYTIDINE'
      na  = 28
      nb =  29
      nhb =  0

c ATOM LABELS

      data (iatom(i),i=1,8)/
     1    'O1''',
     2    'C1''',
     3    'C2''',
     4    'C3''',
     5    'C4''',
     6    'C5''',
     7    'N1  ',
     8    'O3''',
      data (iatom(i),i=9,13)/
     9    'H1''',
     Z    'H2''1',
     1    'H2''2',
     2    'H3''',
     3    'H4'''/
      data (iatom(i),i=14,17)/
     4    'HO3''',
     5    'H5''1',
     6    'H5''2',
     7    'H5''3'/
      data (iatom(i),i=18,22)/
     8    'C2  ',
     9    'N3  ',
     Z    'C4  ',
     1    'C5  ',
     2    'C6  '/
      data (iatom(i),i=23,28)/
     3    'O2  ',
     4    'N4  ',
     5    'HN41',
     6    'HN42',
     7    'H5  ',
     8    'H6  '/

c  ATOM CODES (1-N, 2-C, O-3, P-4, H-5)

      data (id(i),i=1,13)/3,5*2,1,3,5*5/
      data (id(i),i=14,28)/4*5,2,1,3*2,3,1,4*5/

c  BOND ARRAYS
c ipl and ip2 are arrays to specify the bonded atoms.
c Bond m involves atoms ipl(m) and ip2(m).
c ipt is an array to specify the bond type: 1 for single bonds,
c 2 for double bonds, 3 for hydrogen bonds and 4 for aromatic
c bonds.

      data (ipl(i),i=1,13)/2,5,3,7,4,5,8,6,9,10,11,12,13/
      data (ip2(i),i=1,13)/1,1,2,2,3,4,4,5,2, 3, 3, 4, 5/
      data (ipt(i),i=1,13)/13*1/

      data (ipl(i),i=14,29)/14,15,16,17,18,22,19,23,20,21,24,22,
     # 27,28,25,26/
      data (ip2(i),i=14,29)/ 8, 6, 6, 6, 7, 7,18,18,19,20,20,21,
     # 21,22,24,24/
      data (ipt(i),i=14,29)/ 1, 1, 1, 1, 4, 4, 4, 2, 2, 4, 2, 2,
     #  1, 1, 1, 1/

c PARTIAL CHARGES

      data (coul(i),i=1,13)/-0 272,0.308,-0.066,0.187,
     #          0 115,-0 153,-0.164,-0.356,
     #         -0.011, 0.076, 0.000,-0.040,-0.017/
      data (coul(i),i=14,28)/+0 200,0.051,+0.051,0 051,
     #          0 421,-0 348,+0 332,-0.174,
     #         +0.189,-0.419,-0.201,+0.107,0.103,0.020,-0 010/

c  COORDINATES
c Get coordinates for the sugar as a function of the
c pseudorotation phase angle and puckering amplitude
c ("phase" and "puckam"):

      idna = 1
      phase=162.0d0
      puckam = 38.7
```

```
c set up indices for ring atoms O1', C1', C2', C3', C4'
      inring(1)=1
      inring(2)=2
      inring(3)=3
      inring(4)=4
      inring(5)=5

c set up indices for substituents of ring atoms O1', C1', C2', C3', C4'.
c INEXO(I,J), (I=1,5, J=1,2)  specifies the Jth substituent of ring atom I.
c The order for the 2 substituents is such that the 2 atoms are listed
c according to (-tetrahedral) and then  (+tetrahedral)  orientation about
c the ring bond  Thus we have: INEXO(1,1) and INEXO(1,2) - 0 and 0  (no
c substituents to O1'), INEXO(2,1) and INEXO(2,2) - indices of H1' and N;
c INEXO(3,1)  and INEXO(3,2) - indices of 1H2' and 2H2';
c INEXO(4,1)  and INEXO(4,2) - indices of O3' and H3'; and
c INEXO(5,1)  and INEXO(5,2) - indices of H4' and C5'.
c
      inexo(2,1)=  9
      inexo(2,2)=  7
      inexo(3,1)= 10
      inexo(3,2)= 11
      inexo(4,1)=  8
      inexo(4,2)= 12
      inexo(5,1)= 13
      inexo(5,2)=  6

      call cordsu(phase,puckam,delta,xx,maxa,inring,inexo,idna)

c Attach exocyclic  atoms to the sugar ring:
c ------------------------------------------

c attach HO3'.
      call putl(xx,maxa,3,4,8,14,1.0,trig,80.0)
c attach 1 hydrogen of the methyl group of C5':
      call putl(xx,maxa,1,5,6,16,1.0,tet,60.0)
c attach other 2 hydrogens:
      call put2(xx,maxa,5,6,16,15,0.0,+ tet/2.0d0,1.0d0)
      call put2(xx,maxa,5,6,16,17,0.0,- tet/2.0d0,1.0d0)

c Attach Cytosine:
c ----------------

c attach ring atoms C2,N3,C4,C5 and C6:
      call putl(xx,maxa, 1, 2, 7,18,1.39,trig,  ch1)
      call putl(xx,maxa, 2, 7,18,19,1.36,trig,180.0)
      call putl(xx,maxa, 7,18,19,20,1.34,trig,  0.0)
      call putl(xx,maxa,18,19,20,21,1.43,trig,  0.0)
      call putl(xx,maxa,19,20,21,22,1.36,trig,  0.0)
c attach O2
      call putl(xx,maxa, 2, 7,18,23,1.24,trig,  0.0)
c attach N4
      call putl(xx,maxa,18,19,20,24,1.32,trig,180.0)
c attach 2 hydrogens of N4
      call putl(xx,maxa,19,20,24,25,1.0,trig,180.0)
      call put2(xx,maxa,20,24,25,26,0.0d0,0.0d0,1.0d0)
c attach H5 and H6
       call putl(xx,maxa,19,20,21,27,1.0,trig,180.0)
       call putl(xx,maxa,20,21,22,28,1.0,trig,180.0)

      return
      end

c*****************************************************************
```

```
c ***************************************************************
      subroutine  growsu(z, nz, igp, lkz, var1, var2, zlge, zsml)
c
c This routine prepares an array ZSML(3,3) - coordinates of the three
c phosphate atoms P,O1,O2 - from a given sugar array  when geometrical
c variables describing the sugar to phosphate orientation are specified.
c This array is required at the moment only for linking  a sugar with the
c O5' phosphate end (via routine MATCH), LINK 2 as indicated below.
c The routine can be easily extended if necessary to provide an array for
c linking with the O3' phosphate end (link 3) or the base (link 1).
c The variables used are along the same lines as in GROWCG:
c  Z(NZ,3)     - coord. array sugar unit.
c  IGP         - group to be linked with sugar:
c                Group 1 - with base, Group 2 - with O5' phosphate, and
c                Group 3 - with O3' phosphate.
c  LKZ         - index of sugar linker atom.
c  VAR1        - for group 2 - backbone dihedral angle C4'-C5'-O5'-P (beta)
c  VAR2        - for group 2 - backbone dihedral angle C5'-O5'-P-O3' (alpha)
c  ZSML        - array which on output supplies coords. for linking
c                For group 2, the atoms are P, O1, O2, in this order.
c---------------------------------------------------------------
c                  O1
c                  /
c                 /
c      O2 ---- P
c               *  <<<<  LINK 2 (O5' to P of O5')
c               |
c       (#15)   O5'        O1'       N
c                /         /         \     *   <<<<  LINK 1 (C1' to N of base)
c            (#12)  (#10)              \   *
c       2H5'---C5'---C4'                C1' (#2)
c              /      /  \H3' 2H2'/ \
c          H5'    H4'    \|    |/   H1'
c                          C3'---C2'
c                         /       \
c                        /         \
c                (#8)  O3'          1H2'
c                  *
c                   *  <<<<  LINK 3  (O3' to P of O3' end)
c                  P
c---------------------------------------------------------------
c                  LINK 2
c                  ---------
c LKZ  |          15 (index of O5')
c IX1  |          10 (index of C4')
c IX2  |          12 (index of C5')
c IX3  |          15 (index of O5')
c BL1  |           O5'-  P
c BL2  |           P   - O1
c BL3  |           P   - O2
c BA1  |         C5' - O5' - P
c BA2  |         O5' - P   - O1
c BA3  |         O5' - P   - O2
c DA1  |       C4' - C5' - O5' - P    (beta)
c DA2  |       C5' - O5' - P   - O1   (related to alpha)
c DA3  |       C5' - O5' - P   - O2   (related to alpha)
c---------------------------------------------------------------
      implicit double precision(a-h,o-z)
      parameter(nasu=15)
      common/cordsu/xxsu(nasu,3)
      double precision z(nz,3), zlge(nz+3,3), zsml(3,3)
      integer igp, lkz, ix1, ix2, ix3
      double precision var1, var2
      double precision bl1,bl2,bl3, ba1,ba2,ba3, da1,da2,da3
      double precision tet, baback, baPO1, baPO2, dadif1, dadif2
      tet= 109.4712206344907 d0

      goto (1,2,3), igp
c  ------------------
    1 continue
            lkz =  2
      goto 10
c  ------------------
    2 continue
            lkz = 15
            ix1 = 10
            ix2 = 12
            ix3 = 15
c DA2 is related to alpha by the bond angles  BAPO1: O3'-P-O1,
c     BA2: O5'-P-O1,  and  BABACK: O5'-P-O3'.
c DA3 is similarly related to alpha by bond angles  BAPO2: O3'-P-O2,
c     BA3: O5'-P-O2,  and  BABACK: O5'-P-O3'. SEE DIAGRAM BELOW:
c
c   C5'                  O3'          C5'                  O3'
c     \                 /               \                 /
c      \   ___________  /                 \   ___________  /
c       \  |BABACK  /  \                   \  |BABACK  /  \
c        O5'------- P  BAPO1 )             O5'------- P  BAPO2 )
c            | BA2   \    /                   | BA3   \    /
c            --------\  /                     --------\  /
c                     \/                               \/
c                     O1                               O2
c
c Specifically,  alpha = DA2  = +- DADIF1,  and
c                alpha = DA3  = -+ DADIF2, where DADIF1 and DADIF2
c                are expressed below
c If we assign bond angle values from phosphate diesters models,
c        BA2 = BA3 = 108, BABACK = 103, BAPO1 = BAPO2 = 108, then
c        DADIF1  = DADIF2 = 114.109 degrees
c Thus we can use  DA2 = alpha - DADIF1 and DA3 = alpha + DADIF2
c For other desired values for these angles, compute DADIF1, DADIF2.

            bl1 =   1.6   d0
            bl2 =   1.48  d0
            bl3 =   1.48  d0
            ba1 = 120.0   d0
            ba2 = 108.0   d0
            ba3 = 108.0   d0
            da1 =  var1
         baback = 103.0 d0
         baPO1  =  108.0 d0
         baPO2  =  108.0 d0
         dadif1 = dacosd ( (dcosd(baPO1) - dcosd(ba2) * dcosd(baback) ) /
        1                                (dsind(ba2) * dsind(baback)) )
         dadif2 = dacosd ( (dcosd(baPO2) - dcosd(ba3) * dcosd(baback) ) /
        1                                (dsind(ba3) * dsind(baback)) )
            da2 =  var2 - dadif1
            da3 =  var2 + dadif2
      goto 10
c  ------------------
    3 continue
            lkz =  8
      goto 10
c  ------------------
   10 continue
c  ------------------
      call apend3 ( z, nz, zlge, zsml, ix1,ix2,ix3,
     1                  bl1,bl2,bl3, ba1,ba2,ba3, da1,da2,da3)

      return
      end
c ***************************************************************
```

NYU COMPSCI TR-348 c.2
Schlick, Tamar
A modular strategy for
generating starting
conformations and data

NYU COMPSCI TR-348 c.2
Schlick, Tamar
A modular strategy for
generating starting
conformations and data

This book may be kept MAY 04 1988
FOURTEEN DAYS
A fine will be charged for each day the book is kept overtime.

GAYLORD 142 PRINTED IN U.S.A.